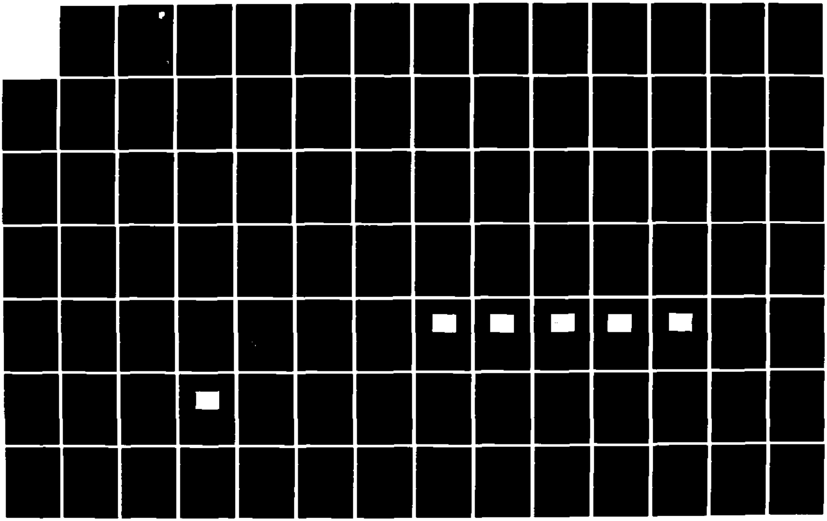
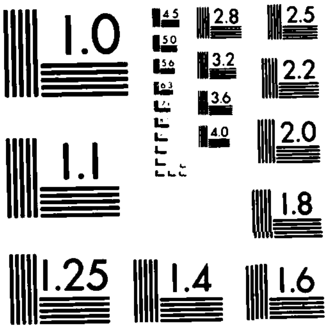
AD-A123 445   SAMPLE DATA PROCESSING(U) MITRE CORP BEDFORD MA   1/2
D O CARHOUN ET AL. AUG 82 MTR-8699 RADC-TR-82-227
F19628-81-C-0001

UNCLASSIFIED   F/G 17/2   NL




1.0
4.5
5.0
5.6
6.3
7.1
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

12

RADC-TR-82-227
Final Technical Report
August 1982

SAMPLED DATA PROCESSING

The MITRE Corporation

D. O. Carhoun, R. J. Cosentino and S. J. Meehan

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, NY 13441

This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-82-227 has been reviewed and is approved for publication.

APPROVED: Frederick D. Schmandt

FREDERICK D. SCHMANDT
Project Engineer

APPROVED: B Beek

BRUNO BEEK, Technical Director
Communications Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (DCCR) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document requires that it be returned.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER RADC-TR-82-227	2. GOVT ACCESSION NO. AD-A123 445	3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
SAMPLE DATA PROCESSING

5. TYPE OF REPORT & PERIOD COVERED
Final Technical Report
Sep 80 - Oct 81

6. PERFORMING ORG. REPORT NUMBER
MTR 8699

7. AUTHOR(s)
D. O. Carhoun
R. J. Cosentino
S. J. Meehan

8. CONTRACT OR GRANT NUMBER(s)
F19628-81-C-0001

9. PERFORMING ORGANIZATION NAME AND ADDRESS
The MITRE Corporation
P. O. Box 208
Bedford MA 01730

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
62702F
MOIE7090

11. CONTROLLING OFFICE NAME AND ADDRESS
ESD/TOF
Hanscom AFB MA 01731

12. REPORT DATE
August 1982

13. NUMBER OF PAGES
134

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)
Rome Air Development Center (DCCR)
Griffiss AFB NY 13441

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Frederick D. Schmandt (DCCR)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Modem
Secure Voice
Filters
Finite Field Algebra

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This MITRE Technical Report is the final report for fiscal 1981 on MITRE MOIE Project 7090: Sampled Data Processing. A general objective of the project was to develop and assess the impact of finite algebraic methods on the signal and data processing functions encountered in command, control, and communications systems. The work in FY81 focused on two areas: development and evaluation of a simple technique for secure transmission of voice-band signals over voice-grade telephone lines and concurrently,

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

comparative design of finite field and conventional digital filters. Work in both areas is covered in this final report.

Department Approval: Ronald D. Haggarty

MITRE Project Approval: Dean O. Carhoun

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONCLUDED)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (CONTINUED)

LIST OF ILLUSTRATIONS (CONCLUDED)

LIST OF TABLES

LIST OF TABLES (CONCLUDED)

SECTION I

INTRODUCTION

1.1 OVERVIEW

This MITRE technical report is the final technical report for fiscal 1981 on MITRE MOIE Project 7090: Sampled Data Processing. A general objective of the project was to develop and assess the impact of finite algebraic methods on the signal and data processing functions encountered in command, control and communications systems. It is a premise of the work that the practical effects of numerical constraints on the realization of processing algorithms can be alleviated by working in finite mathematical structures with their attendant benefits: reduced roundoff error accumulation, stable designs for recursive structures, parallel architecture for integrated electronics implementation, enhanced testability of resultant hardware, and simple application of systematic design procedures. The work in FY81 focused on two areas: development and evaluation of a simple technique for secure transmission of voice-band signals over voice-grade telephone lines and, concurrently, comparative design of generic digital filtering processors by finite algebraic and conventional binary means. The project carried out work previously initiated under the Sampled Data Processing Techniques IR&D project.

1.1.1 Accomplishments

Progress was made in practical development of a simple method of secure analog voice transmission, that does not require source coding or channel bandwidth expansion, based on a technique of masking the analog voice samples with a pseudorandom sequence by addition in a ring of integers. We completed a conceptual design of the modem, and devised several practical methods of contending with the time-dispersive channel. Alternative approaches, described in a MITRE report [1], were compared. A hardware simulation was performed to compare the relative practicality of compensating the channel with an approach of predistorting the masking sequence, by processing in a filter that replicates the channel response, with a conventional approach of equalizing the channel with an inverse filter. The predistortion method demonstrated a cancellation ratio of 42 dB, for a pseudorandom sequence transmitted through a commercial telephone line simulator, by means of a simple tapped delay line channel replicating filter. The results of this phase of the work will be presented in a forthcoming paper at the 1982 IEEE-sponsored Carnahan Conference on Security Technology.

Progress was also made in comparative design of generic digital filters, some motivated by the secure voice modem. The examples described in this report include a lowpass filter designed to avoid intersymbol interference, an interpolation filter required for changes in digital sampling rates, and the telephone channel replicating and compensating filters. We concentrated on direct implementation of the convolution function by means of computationally efficient hardware suitable for integrated electronics and related to precise recursive design algorithms. This is a significant departure from the present trend toward computing digital convolutions by fast transform algorithms. Simple software design tools were developed for operation in finite fields. They were used for the purpose of finite field simulation of the design examples contained in this report.

1.2 MAJOR WORK AREAS

1.2.1 Secure Voice Bandwidth Modem

The Sampled Data Processing project continued efforts begun in

1980 under MITRE IR&D sponsorship that were associated with the development of a technique for secure voice-band analog transmission that could employ finite algebraic digital processing for design and implementation of the high-precision filter functions required by the modem. Techniques examined in fiscal 1980 were breadboarded in FY81, the breadboard modem was tested and evaluated, the relevance and signal processing potential of the finite algebraic methods established, and the results disseminated [4].

During fiscal 1981 this novel technique for secure voice-band analog transmission was developed and evaluated by hardware simulation. The resulting design provided digital filter requirements for lowpass, interpolation and equalization filters that provided examples for comparative evaluation of the conventional and finite algebraic design approaches.

Section II of this report provides an in-depth discussion of the design aspects of the modem and presents results of an experiment, based on hardware simulation, of a critical developmental aspect of the modem: removal of a pseudo-random sequence, used for data encryption by sample-value masking, after transmission through a time-dispersive channel that creates intersymbol interference. Feasibility of the modem concept requires that a simple means of channel compensation be implemented to cancel the interference.

Two methods of compensation were examined. One method used the conventional approach of compensating the channel with an inverse filter. The other method, which relies on the essential linearity of the channel, attempts to replicate the measured channel response for the purpose of prefiltering the reference sequence at the receiver prior to cancellation of the reference component from the masked sequence.

Implementation of the modem requires both digital and analog processing functions since the transmission medium supports sampled analog data. The critical processing functions, however, are implemented digitally. They provided design examples for evaluation of finite field digital filtering techniques.

1.2.2 Finite Algebraic Signal Processing

The Sampled Data Processing project is a continuing, comprehensive effort aimed at improved techniques of design and implementation of linear signal processing functions for communication and sensor systems. Traditional methods of digital signal processing suffer from compromises and approximations resulting from quantization and roundoff imposed by the numerical constraints of binary representations of continuous signals. This project is aimed at fundamentally new, improved design and implementation capabilities based on finite algebraic techniques.

Practical utilization of the fundamental algebraic properties of discrete linear systems can reduce the numerical constraints on conventional digital signal processors. Benefits can be realized as reduced roundoff error effects, faster throughput produced by fast computational algorithms and reinforced by parallel architecture, simpler hardware structures, improved stability for recursive processors, and streamlined, systematic design procedures.

Technological advances make it increasingly attractive to implement most signal processing functions digitally. While digital equipment is limited to a finite number of digits of calculation accuracy, the signals being processed generally have no such theoretical limit, and the required calculations are defined for a continuum of values based on normal arithmetic capable of producing a number of any size. The conventional design of digital processing systems typically results in a series of compromises and approximations emanating from the need for finite quantization and a limited range of numbers at each operation. These compromises can lead to hardware and software implementations which are cumbersome and potentially unreliable when implemented on a very large scale. The problem is compounded by the necessity to thoroughly test the equipment to verify successful and reliable operation, since the effects of roundoff error accumulation are usually intractable, and at best difficult to prevent, analyze or predict.

Traditional design approaches and implementation methods are being actively challenged by new, fundamentally digital, techniques rooted in finite algebra and made feasible by large-scale integrated circuits. In the mathematics of finite algebra the defined operations are closed, encompassing an invariable and finite set of values. All calculations are exact and answers are always contained in the invariable set of exact values, preventing compromises and approximations during computation.

The advantages are realized in the hardware implementation of the digital processor. For example, one may be faced with the requirement to implement a digital filter having many values in the impulse response, to a precision of many decimal digits and exhibiting a large ratio of maximum to minimum non-zero value,in order to accurately produce a desired response in either the time domain or frequency domain. Binary (radix two) arithmetic in the processor would require many bits to represent each value and the difficulty would be reflected chiefly in the circuit complexity of the multipliers. The tendency is to use floating-point arithmetic and recurrent operation of a few critical components (multipliers, accumulators) to implement the function. The result is to reduce the throughput and increase the processing delay compared to that available in a fully parallel processor.

Alternatively, the finite algebraic approach allows the processor to be represented in a finite ring, large enough to encompass the required dynamic range, and capable of being decomposed into a parallel set of processors operating over finite fields of relatively prime characteristic whose external direct product represents the original finite ring. The principal advantage is that the individual processors can make use of simpler functional elements (multipliers, accumulators) that can use a modest number of bits in equivalent fixed-point arithmetic and be configured with a high degree of circuit regularity that lends itself to modern digital integrated circuitry.

The essential structure of a finite field processor is shown schematically in Figure 1. Implementation requires that the ring modulus be large enough for the processing problem at hand so that overflow errors do not occur and so that the rounding error that results from mapping the input samples and filter coefficients, ordinarily occurring as numbers in the field of rationals, into a finite ring can be contained within the required limits. After this small error has been accepted, no further error results from the finite field computation. The modulus must also be a composite number that can be factored into the product of different primes.

Section III of this report presents the results of designing and using several digital filters in a finite ring that is decompossed into the direct sum of prime fields. The examples were chosen from the modem hardware described in Section II. Appendix A contains a BASIC program listing of one of the software routines used to perform the filtering computations.

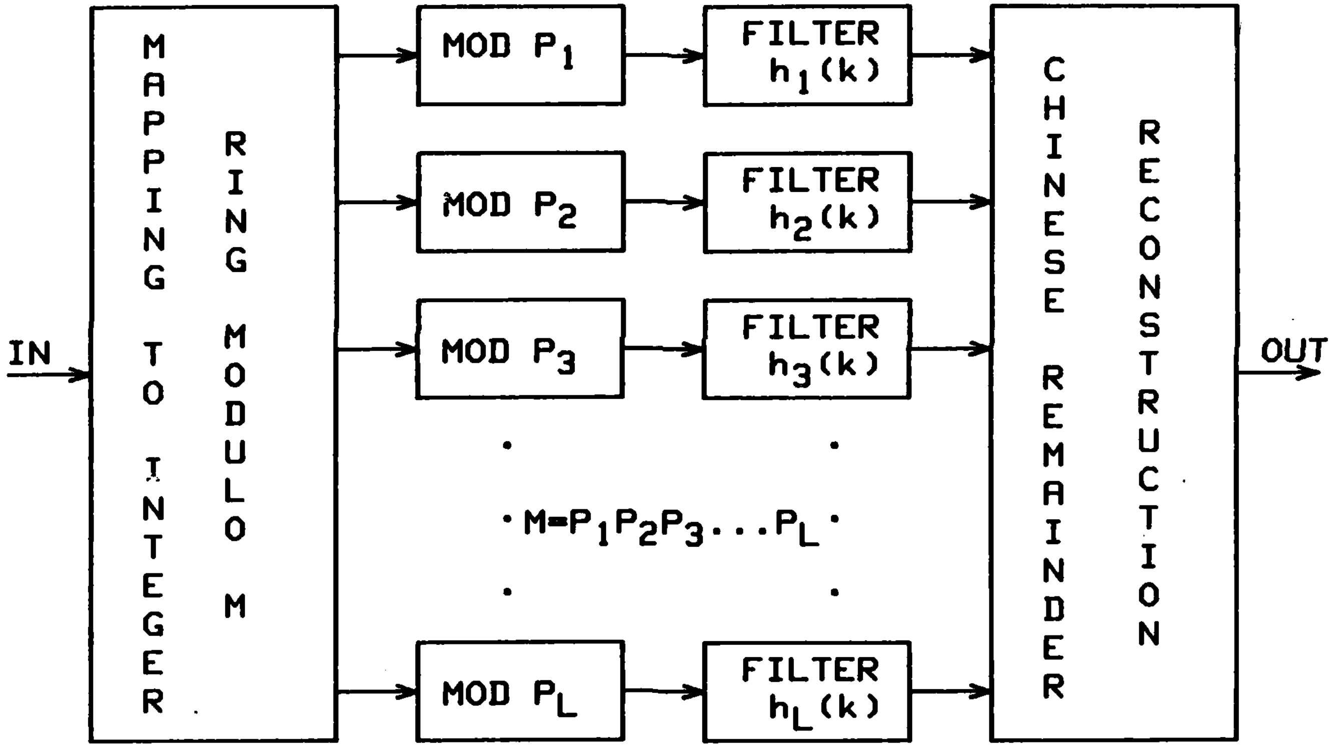


Figure 1. FINITE FIELD PROCESSOR

1.2.3 Conclusions and Future Work

The final section of this report, Section IV, presents some conclusions relevant both to the secure-voice bandwidth modem and to the methods of finite-field digital filtering. It also outlines future developmental work needed in both areas.

SECTION II

SECURE VOICE-BANDWIDTH MODEM

2.1 BACKGROUND

2.1.1 Overview

Speech encryption can protect against unauthorized access to conversations being transmitted over telephone lines. Analog encryption devices do not offer the kind of security required for many military (and business) applications. While digital devices are secure, they generally require bandwidths much wider than provided by voice-grade telephone lines or they require complex voice compression techniques. This section describes an attempt to design a method of transmitting encyphered (digitally encrypted) speech over voice-grade analog telephone lines with relatively simple and inexpensive equipment and with little degradation in the quality of the decyphered speech. It also presents the results of an experiment that embodies, in part, the design described and that tests critical aspects of design feasibility.

Two types of encryption are commonly available--speech scrambling and speech cyphering. In scrambling, voice is divided into small time-bandwidth sections and transformed or rearranged according to some program to eliminate the intelligibility to the human ear. This type of encryption is characteristically less complex and less expensive to implement than speech cyphering, but it is less secure because of the residual clear-voice information in the cryptogram. Speech cyphering, in which digitized voice is modified by modulus addition (usually modulo 2) with a pseudo random sequence from a cyphering generator, can be made much more secure, but it requires complex and expensive equipment to obtain the bandwidth required for clear-voice transmission over voice-grade telephone lines and also provide for accurate decryption to retain clear voice quality at the receiver.

A major problem encountered in the transmission of 2700 Hz wide encryption signals is the Nyquist filter design to be used in the telephone system. Intersymbol interference is severe for a 2700 Hz bandwidth signal sent through a relatively good voice-grade C2 line, as indicated by the group-delay curve in Figure 2.

To avoid intersymbol interference, Nyquist filtering is added to the telephone line. The filter provides an impulse response with zeros at intervals that are multiples of the sampling interval T. As illustrated in Figure 3, the "channel" includes everything in the system between the output of the digital-to-analog converter (B) and the input of the analog-to-digital converter (C). If the sampling times at the analog-to-digital converter (ADC) are synchronized to the signal so that the converter samples at the nulls (of all but one) of the impulse responses, then the converter samples the magnitude of the input impulse corresponding to that sampling time. Ideally the numbers at Point D are identical to those at Point A.

The problem arises because the bandwidth of the Nyquist filter in this case is almost as wide as the bandwidth of the telephone line. The usual practice when transmitting digital data through a telephone line is to choose a baud rate low enough to keep the bandwidth of the Nyquist filter significantly narrower than the telephone channel bandwidth. The Nyquist filter determines the resultant frequency response, i.e., over that region in the frequency domain where the signal power out of the Nyquist filter is significant, the telephone line has flat amplitude response and a linear phase response so that it contributes only to the phase delay and not to any phase distortion of the signal. Any effect

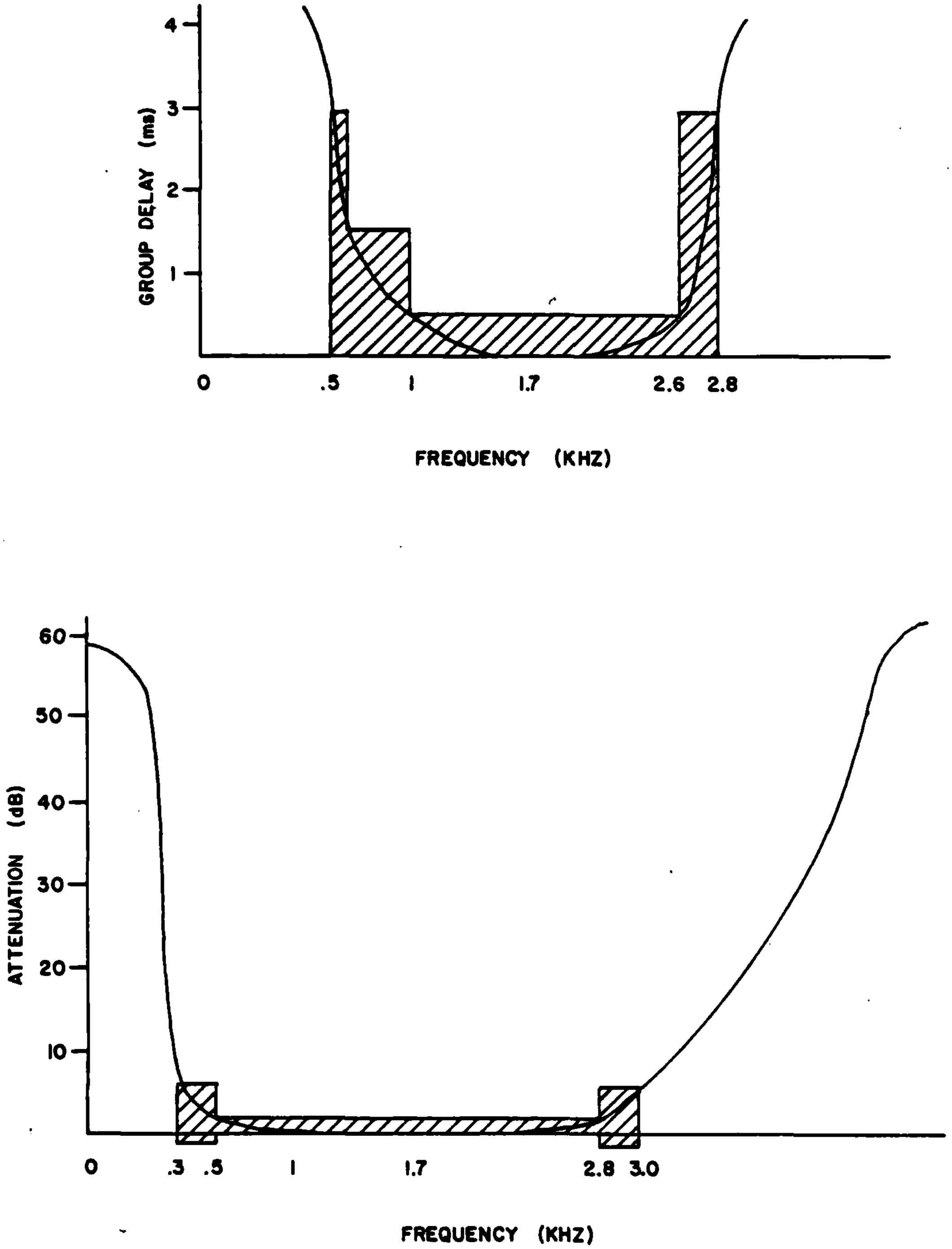


Figure 2. GRAPHICAL REPRESENTATION OF BELL SYSTEM SCHEDULE 3002

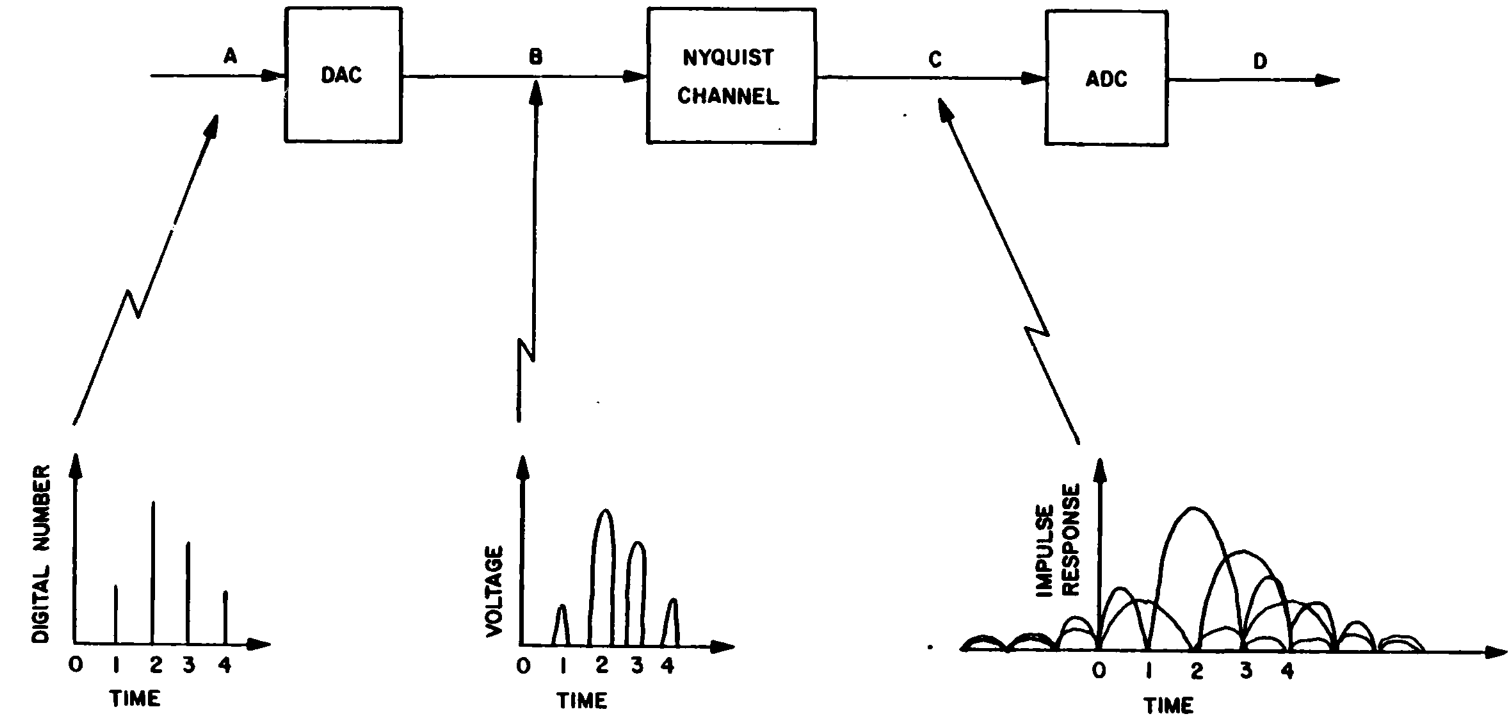


Figure 3. IMPULSE RESPONSE IN A NYQUIST CHANNEL

on the total response by the telephone line becomes a "second order" effect. But when the telephone line has a bandwidth close to that of the Nyquist filter bandwidth, the telephone line characteristics have a "first-order" effect on the total frequency response of the system. What makes this problem particularly difficult to assess is the fact that the characteristics of the telephone lines vary: not only do the characteristics of one line differ from another line, but on any one line they vary with time. However, this report will not be concerned with specific techniques for adapting to time-varying channel characteristics, but will attempt first to examine the feasibility of the stationary case.

2.1.2 Design Features

The encyphering process described here uses a "sample masking" technique: a sequence of eight-bit pseudo-randomly chosen integers is added modulo 256 to an eight-bit per sample digital voice sequence in a 5400 Hz bandwidth (unlike, for example, modulo 2 addition which spreads the bandwidth eight-fold). The modulus addition of voice and encryption sequences may be equivalently thought of as the ordinary addition of the sequences: a voice sequence, $\{V_n\}$; an encryption sequence, $\{E_n\}$; and a modulus sequence, $\{M_n\}$. The values of the voice terms and the encryption terms are integers in the interval from zero to 255 inclusive. The modulus terms each have one of two values determined from the other two sequences: $M_n = 0$, if $V_n + E_n < 256$; $M_n = -256$, if $V_n + E_n \geq 256$.

A single-channel scheme is used instead of in-phase and quadrature channels. The use of cross-coupled equalizers required for the two-channel scheme is thereby avoided, and the total number of circuits is decreased.

Instead of a channel equalizer, a channel replicator compensates for channel distortion. The replicator is expected to remove more of the encryption component from the received signal than the equalizer can, but the comparison of compensation methods is the subject of experimentation.

The receiver determines the modulus sequence from the received sequence (the sequence constructed from the received analog signal) by first extracting the encryption component, leaving the voice and modulus components. If the received signal is undistorted by the channel, the modulus terms can then be retrieved from the remaining sequence simply and with no special processing modifications.* If, however, consideration must be given to channel distortion, in order to mitigate the errors in retrieving the modulus sequence due to the distortion, the single-channel configuration unlike the two-channel configuration may allow the exploitation of the correlation existing between adjacent voice samples to neutralize the error-causing effect of the channel distortion.

2.1.3 Experimental Performance

A single-channel configuration using the sample-masking algorithm was built to test the critical aspects of the scheme and to compare the relative capabilities of the channel replicator and channel equalizer configurations to cancel an encryption sequence distorted by the channel. Using a 61-tap transversal filter as a channel replicator, the replicator configuration was able to cancel the

* A simple algorithm for retrieving the modulus terms from an undistorted sequence is:

$$\text{Decide } M_n = 0 \quad , \text{ if } V_n + M_n \geq 0$$

$$M_n = -256, \text{ otherwise.}$$

received encryption sequence to the extent that the residue out of the subtractor was 42 dB below the encryption sequence entering the subtractor. Using a 61-tap transversal filter as a channel equalizer, the residue in the equalizer configuration was down by 32.6 dB.

2.2 SINGLE-CHANNEL CONFIGURATION

A single-channel configuration has two main advantages over one employing in-phase and quadrature channels:

- it is simpler to implement, since it needs fewer circuits and does not require cross-coupled equalizers in the receiver, and
- unlike quadrature channel systems, which mix high-and low-frequency components, the low-frequency components are kept separate from the high-frequency voice components.

Since most of the power in a voice spectrum, in general, is concentrated in the low frequencies, there is some correlation between adjacent samples. Advantage can be taken of this correlation in reconstructing the modulus sequence, instead of providing a "forbidden zone"* which restricts the dynamic range of the voice samples (see Figure 4).

*The concept of the "forbidden zone" is to restrict the range of the signal in the transmitter to some fraction, say three-fourths, of the 256 levels allowed by the 8-bit values. If the total distortion and noise do not change the value of the received sample from that of the transmitted sample by more than 12½% of the full dynamic range (32 levels), the modulus component can be recovered correctly by subtracting out the encryption component from the digital sequence in the receiver and applying the following decision rule to each term of the remaining voice-plus-modulus sequence: Decide

$$M_i = \begin{cases} 0 & \text{if } V_i + M_i > -32 \\ -256 & \text{otherwise} \end{cases}$$

where V_i is the voice component of the term, and M_i is the modulus component.

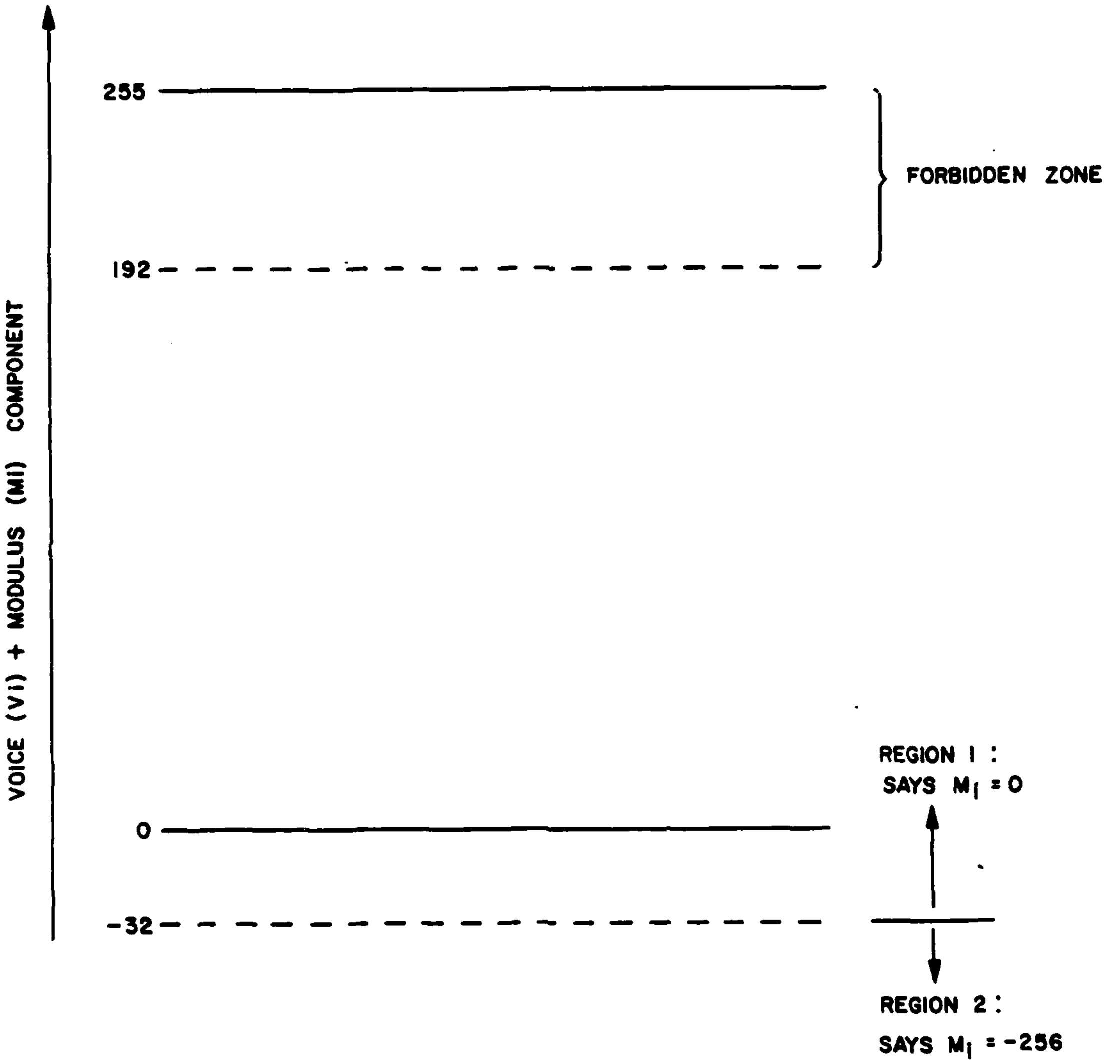


Figure 4. DECISION SPACE FOR THE "FORBIDDEN ZONE" SCHEME

2.2.1 Transmission

The development of this configuration assumes that the channel is linear and stationary and that the voice frequencies to be processed lie in the interval between 300 Hz and 3000 Hz. The analog input waveform is first oversampled at a 10.8 kHz rate to prevent aliasing (see Figure 5). The samples are then digitized and mixed with a 3000 Hz sinusoidal wave resulting in an inverted spectrum. The spectra before and after mixing are sketched in Figures 6a and 6b. After the signal has passed through the low-pass filter (with a transition region between 2700 Hz and 3300 Hz) following the mixer (see Figure 5), only one sideband remains (Figure 6c). Because the signal has not been mixed with its 1650 Hz center frequency to "baseband", there is no overlapping of sidebands. The necessity of using quadrature channels is thus avoided.

In order to pass the encryption stream through the channel with as little distortion as possible, we restrict the bandwidth of the encryption sequence to the 2700 Hz voice bandwidth. To do this we add the encryption sequence to the voice sequence at a 5400 Hz rate. Since the voice spectrum is nonzero only from -2700 Hz to +2700 Hz, we can ignore every other sample of the oversampled 10.8 kHz voice sequence, leaving a 5400 Hz voice sample rate which retains all the information of the original waveform and can be added to the encryption and modulus sequences at the 5.4 kHz Nyquist rate.

The spectrum of the 5400 Hz sample-rate voice signal before encryption is sketched in Figure 6d. The principal, or zero-order, spectrum (the spectrum between -2700 Hz and +2700 Hz) is displaced from the original zero-order spectrum by 2700 Hz, i.e., by mixing the 5400 Hz equivalent voice samples with a 2700 Hz cosine wave we obtain the inverted zero-order spectrum shown in Figure 6e. Since for samples at a 5400 Hz rate,

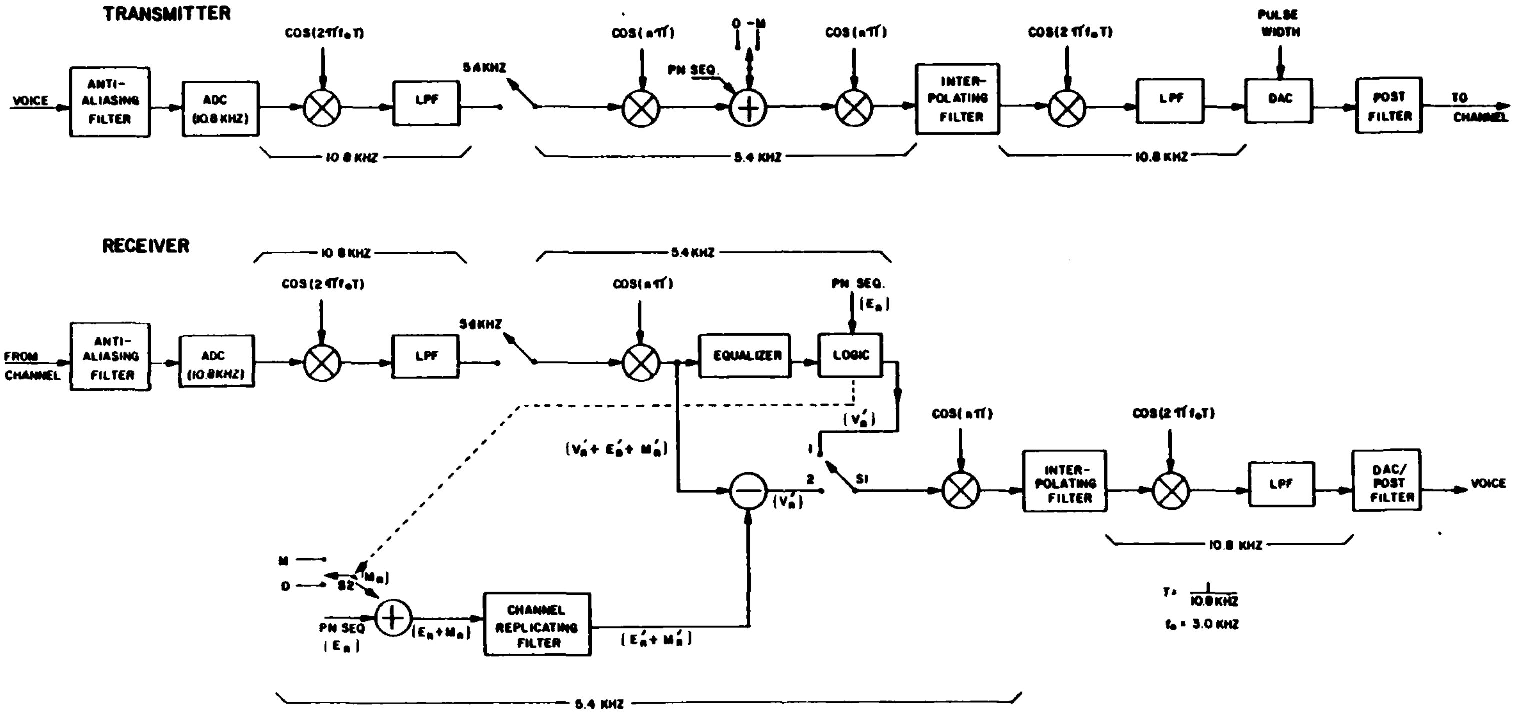
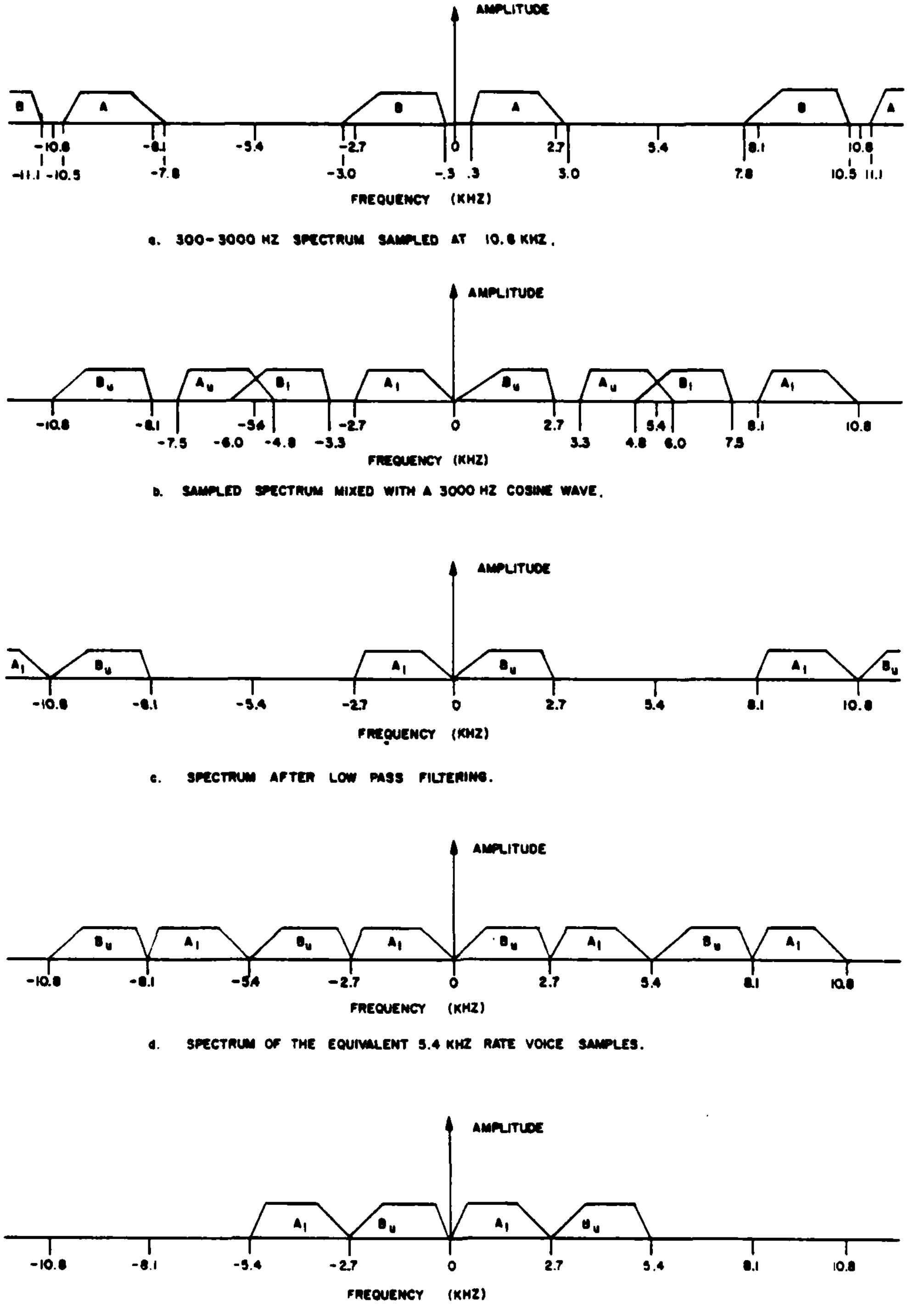


Figure 5. SINGLE CHANNEL CONFIGURATION



a. 300-3000 HZ SPECTRUM SAMPLED AT 10.8 KHZ.

b. SAMPLED SPECTRUM MIXED WITH A 3000 HZ COSINE WAVE.

c. SPECTRUM AFTER LOW PASS FILTERING.

d. SPECTRUM OF THE EQUIVALENT 5.4 KHZ RATE VOICE SAMPLES.

e. SPECTRUM OF THE 5.4 KHZ RATE VOICE SAMPLES AFTER A 2.7KHZ TRANSLATION.

Figure 6. SPECTRA IN THE TRANSMITTER OF THE SINGLE-CHANNEL CONFIGURATION (WITHOUT ENCRYPTION)

$$\cos(2\pi \times 2700t) = \cos(2\pi \times 2700 \frac{n}{5400}) = \cos(\pi n) = (-1)^n$$

the mixing operation is equivalent to multiplying alternately by +1 and -1 both before and after encrypting the voice stream (see Figure 5). The first multiplication allows the addition of the encryption sequence to the original sampled waveform; the second multiplication returns the spectrum of Figure 6d. The 10.8 kHz rate samples out of the interpolator contain the same voice components as the output of the first 3000 Hz mixer.

The 10.8 kHz sample rate allows filtering the unwanted sidebands easily after mixing to a 300 to 3000 Hz band. The digital signal is then converted to an analog signal, smoothed with a filter, and transmitted through the channel.

2.2.2 Reception

The received signal up to the equalizer is processed in a manner similar to the voice signal in the transmitter up to the addition of the encryption sequence. In the replicator configuration (Switch S_1 of Figure 5 in position 2), the equalizer, which is in a branch circuit used only to determine the modulus sequence, compensates for channel distortion only to the extent required to retrieve the bivalued modulus sequence. Since the possible values of the modulus sequence differ by the full dynamic range of the system (256 levels), the equalization need not be as complete as is needed in the equalization configuration (Switch S_1 of Figure 5 in position 1), which must remove distortion to the extent which allows satisfactory cancellation of the encryption sequence as well as the modulus sequence. The equalizer requirements in the replicator configuration may be relaxed considerably more than those in the equalizer configuration.

The "Logic" circuit, after recovering the modulus sequence, adds this sequence to the encryption sequence (indicated by the dotted line in Figure 5). The output of the adder is distorted by the channel replicating filter so that the value of each term out of the replicating filter is ideally equal to the sum of the encryption and modulus components of the corresponding term in the received sequence. The subtractor output then contains the voice sequence along with any residual components of the encryption sequence not removed by the subtractor.

In the equalizer configuration, the equalizer ideally removes the distortion from the received sequence, so that it is identical to the original transmitted sequence. The logic circuit then removes the encryption component of the received sequence, determines the modulus component in each term of the remaining sequence, and removes that component from each term, leaving the voice component in each term of the sequence along with any residual component due to an imperfect equalizer.

Because the mixer immediately preceding the equalizer restored the voice sequence to its original form in the sense that the lower frequencies of the voice sequence at the equalizer input correspond to the lower frequencies of the original voice, the "Logic" circuit may be able to take advantage of the correlation existing between adjacent voice samples to aid in recovering the modulus terms from the still somewhat distorted received sequence out of the equalizer. The nearer the remaining component, $V'_n + M'_n$, of a term in the received sequence is to zero, after cancelling the encryption component, E'_n, the greater is the probability of erring in deciding the value of M_n. The probability of error may be decreased by using the correlation between samples to aid in making a correct decision. A zone may be chosen consisting of an interval of values around zero, in which the probability of error due to channel distortion is unacceptably high. If a term $V'_n + M'_n$ is within that zone, that value of M_n is chosen which puts the value of V'_n closer to V'_{n-1}.

After the removal of the encryption and modulus components from the received sequence, the voice sequence is passed through a mixer which again shifts its spectrum by 2700 Hz. The mixing operation, multiplying the voice sequence by a 5400 Hz cosine wave, reduces, as before, to changing the sign of alternate samples.

The interpolating filter doubles the sampling rate of its output(to 10.8 kHz)over its input rate and filters out the repetitive part of the voice spectrum (between 2.7 kHz and 8.1 kHz). This prevents aliasing of the voice signal in the mixing operation which follows the interpolating filter. The output of the mixer is passed through a low-pass filter to remove the unwanted sidebands. The digital-voice sequence with a 300 - 3000 Hz pass band is then converted to an analog waveform which ideally duplicates the original analog clear-voice signal at the transmitter input.

2.3 EXPERIMENTAL EQUIPMENT

A critical part of this encyphered speech implementation is the removal of the encryption sequence, which has been distorted by the channel, from the received signal. To the extent that a residue of the encryption component of the received signal remains in the decyphered voice signal, the voice quality is degraded. The experiment described in this report was devised to measure the extent to which the encryption can be removed.

Voice-grade public telephone lines were designed and built for the transmission of analog voice signals to be received by the human ear. Owing to the ear's tolerance of amplitude and phase distortion, the distortion permitted in telephone lines is too great to allow sufficient cancellation of the encryption sequence without correcting or compensating for the distortion.

One way to compensate for the distortion is to pass the received signal through an equalizer to restore the waveshape the signal had before entering the channel. Then the encryption sequence can be removed from the received signal by subtracting from it the encryption stream generated in the receiver(generated identically to the stream in the transmitter).

A second way to compensate for the channel distortion is to predistort the encryption stream in the receiver by means of a filter which replicates the impulse response of the channel. The output of the replicating filter, which is more or less identical to the encryption component of the received signal, is subtracted from the digitized received signal to remove the encryption sequence.

Both approaches, the channel equalizer and the channel replicating filter, use a transversal filter for time domain filtering. Their performances can be compared by measuring the value of the residual encryption sequence in each configuration as a function of the number of taps and of the number of significant digits in the tap weights.

In laboratory measurements, a transversal filter having up to 99 taps available with sixteen-bit tap weights was used both as a channel replicator and as a channel equalizer. The extent to which its output could replicate the encryption and modulus components of a sequence distorted by a simulated Bell schedule 3002 voice-grade telephone line, when configured as a channel-replicating filter,was compared with its ability to remove the distortion from these same components when used as a channel equalizer. The comparison between the two approaches was made by transmitting a sequence containing only encryption components through the simulated telephone line and then measuring the residual sequence after attempting to cancel the received sequence with the sequence reconstructed in the receiver. The basic configuration for each approach is shown in Figure 7. For the experiment an inverse filter was used as the channel equalizer.

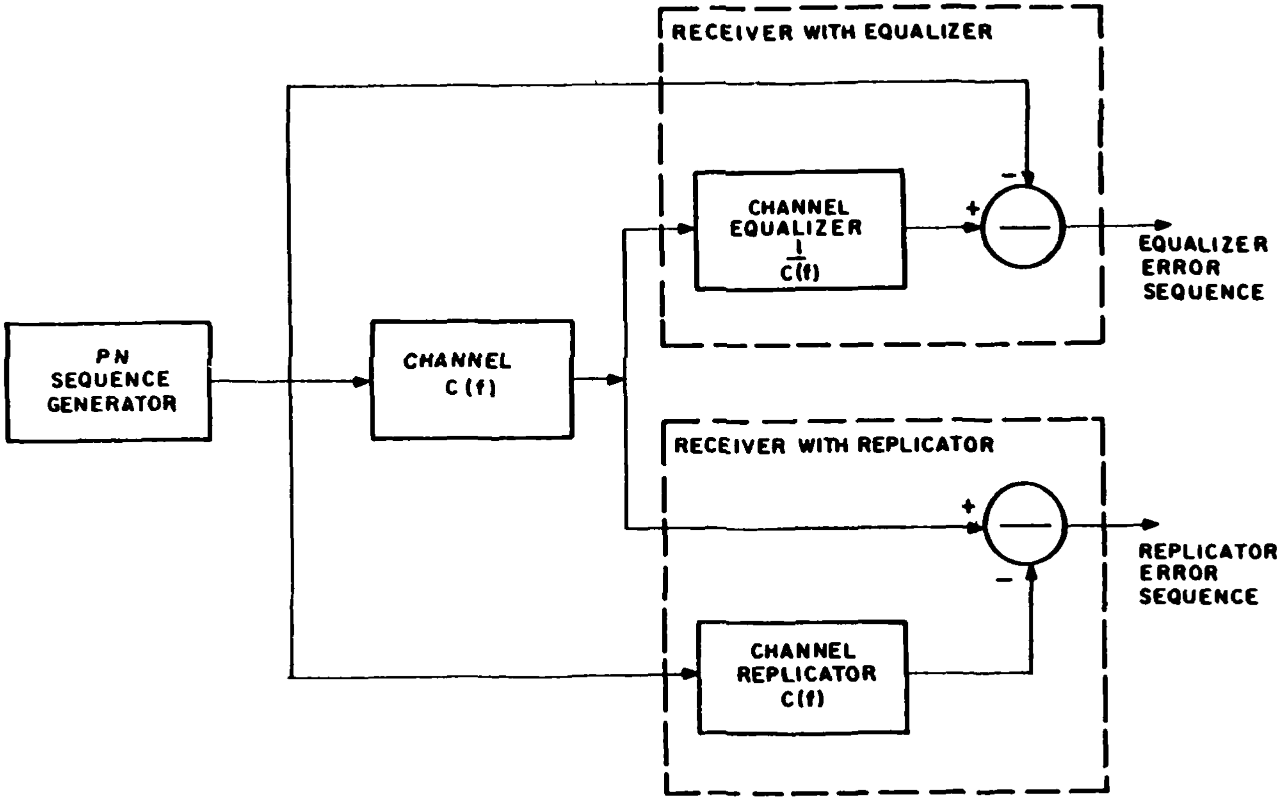


Figure 7. BASIC CONCEPT FOR COMPARING THE CAPABILITIES OF THE CHANNEL EQUALIZER AND CHANNEL REPLICATOR

2.3.1 Channel Configuration

A 2700 Hz-wide digital PN sequence is to be transmitted to the receiver through a simulated telephone line, which is an analog band-pass filter with its pass band extending from approximately 300 to 3000 Hz. Therefore the digital signal must be mixed up to the telephone line pass band and converted to an analog signal before being passed through the telephone line. The receiver, in turn, must convert the received analog signal to a digital signal and mix it back down to its original band in order to reconstruct the original sequence (now somewhat distorted).

It is convenient to include in the "channel" all the circuits between the original sequence at the input to the interpolating filter in the transmitter and the reconstructed sequence at the output of the antialiasing filter in the receiver (Figure 8), since the channel equalizer and the channel replicator must suppress the distortion produced by the included circuits as well as that produced by the telephone line. Those transmitter circuits to be considered as part of the channel are referred to as the "transmitter channel". Those receiver circuits to be considered as part of the channel are referred to as the "receiver channel". The remainder of this section explains the configuration of Figure 8 in detail.

2.3.2 Telephone Line

A SEG Electronics Corporation Model No. FA-1445 telephone line simulator is used to represent the telephone line over the range from 0 to 2700 Hz in the experimental equipment. It is designed, according to the manufacturer, as a "nominal worst case" line, i.e., the simulator incorporates the greatest variation of delay and attenuation that can be fitted on a smooth curve within the ranges specified by the Bell System Schedule 3002. The characteristics of

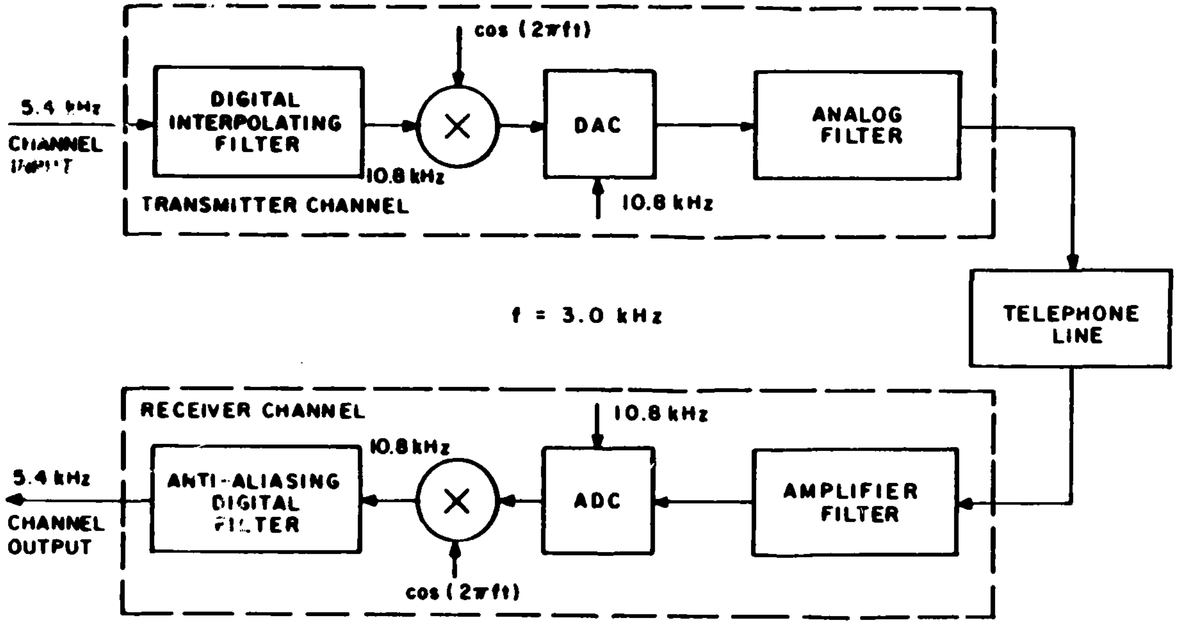


Figure 8. CHANNEL BLOCK DIAGRAM

the simulator given by the manufacturer are listed in Table 1. The ranges were shown graphically in Figure 2. The frequency response of the simulator that we measured in the laboratory is shown in Figure 9.

TABLE 1

CHARACTERISTICS OF THE SIMULATED 3002 TELEPHONE LINE

Insertion Loss*	16 ± 1 dB @ 1000 Hz
Attenuation Characteristic* (Ref. 1000 Hz)	-3 to +12 dB (300-3000 Hz) -2 to +8 dB (500-2500 Hz)
Envelope Delay Distortion (maximum)*	1750 μs (800-2600 Hz)
Above Band Roll-off Characteristic**	80 dB/octave to 50 dB, holding up to 10 kHz
Below Band Roll-off Characteristic**	24 dB/octave to 150 Hz, holding to DC
Envelope Delay at Band Edges (minimum)**	3000 μs

* Bell System Schedule

** "Nominal Worst Case"

The characteristics of the telephone line simulator above 10 kHz are not specified by the manufacturer. Figure 9 shows the attenuation decreasing above 10 kHz to 13 dB at 500 kHz. To correct this, a low-pass filter was added in series with the telephone line to maintain a high attenuation beyond the upper cut-off frequency. The combination of the telephone line simulator and low pass filter, then, forms the telephone line in the experiment.

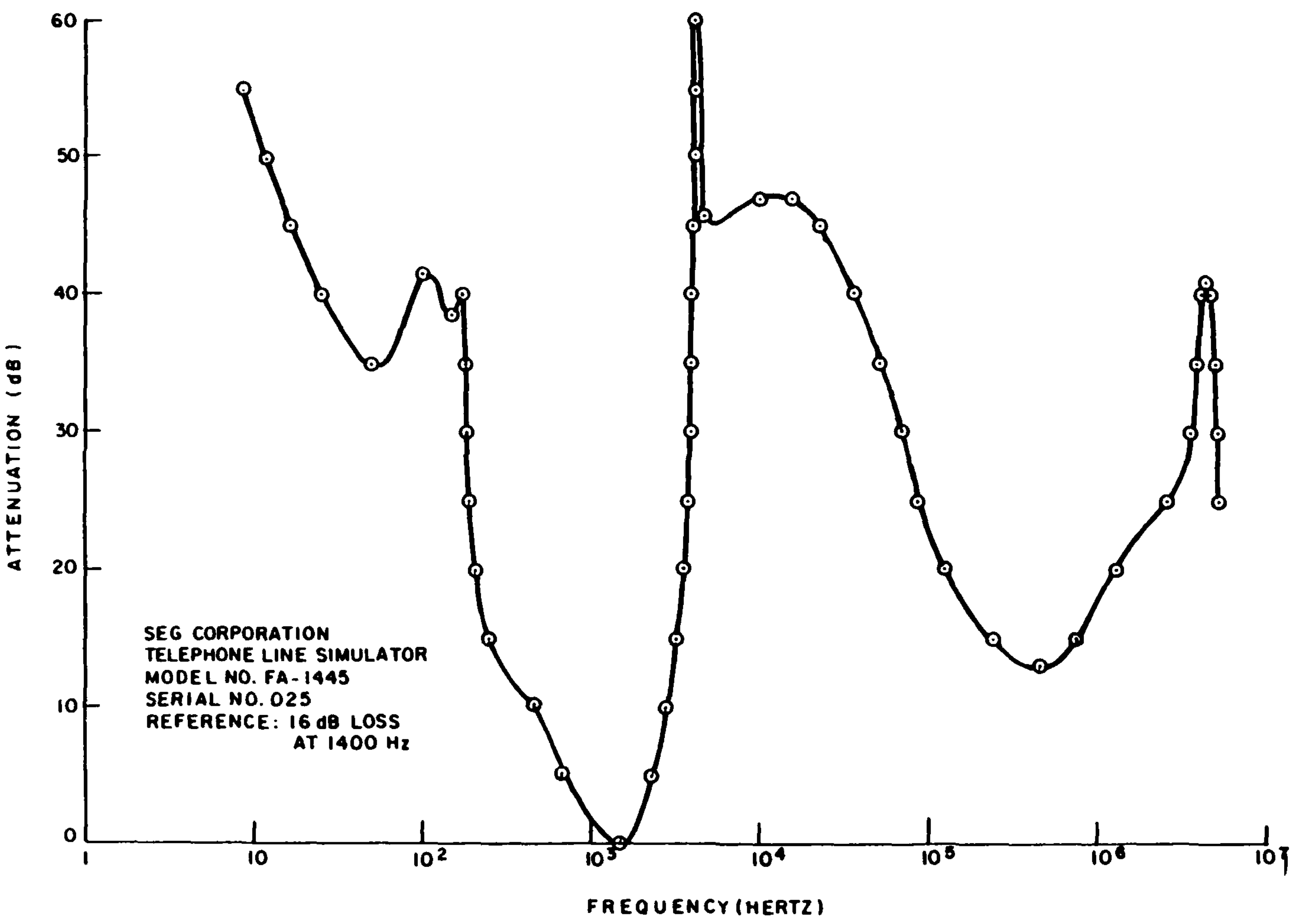


Figure 9. FREQUENCY RESPONSE OF THE TELEPHONE LINE SIMULATOR

2.3.3 Transmitter Channel

The transmitter channel consists of those circuits which convert the 5.4 kHz PN sequence into an analog signal suitable for transmission through the telephone line (Figure 8). The interpolating filter widens the gap between the various orders of the spectrum to prevent overlapping of upper and lower sidebands when the mixer shifts the spectrum of the PN sequence by ± 3.0 kHz. The digital-to-analog converter (DAC) and analog filter complete the conversion of the PN sequence to an analog signal in the frequency band 300 to 3000 Hz.

2.3.3.1 Interpolating Filter. This experiment used a 57-tap transversal filter as an interpolating filter. As a guide in specifying the frequency response of the interpolating filter, a raised cosine (see Figure 10) shape was chosen of the form:

$$x(f) = \begin{cases} 1.0 & f \le 2300 \text{ Hz} \\ \cos \dfrac{2\pi(f-2300)}{1600} & 2300 \text{ Hz} \le f \le 3100 \text{ Hz} \\ 0 & f > 3100 \text{ Hz} \end{cases}$$

The 57 tap transversal filter was designed to approximate the raised cosine frequency response (Figure 11). The most important criterion in designing the filter is that, to avoid intersymbol interference, its impulse response for every odd sample must be zero except the middle sample (29th in this case). The impulse response for the interpolating filter used in this experiment, shown in Figure 12, satisfies this criterion. The tap weights for this filter are listed in Table 2.

2.3.3.2 Mixers. The input to the mixer consists of a set of 16-bit binary numbers exiting the interpolating filter at a 10.8 kHz rate

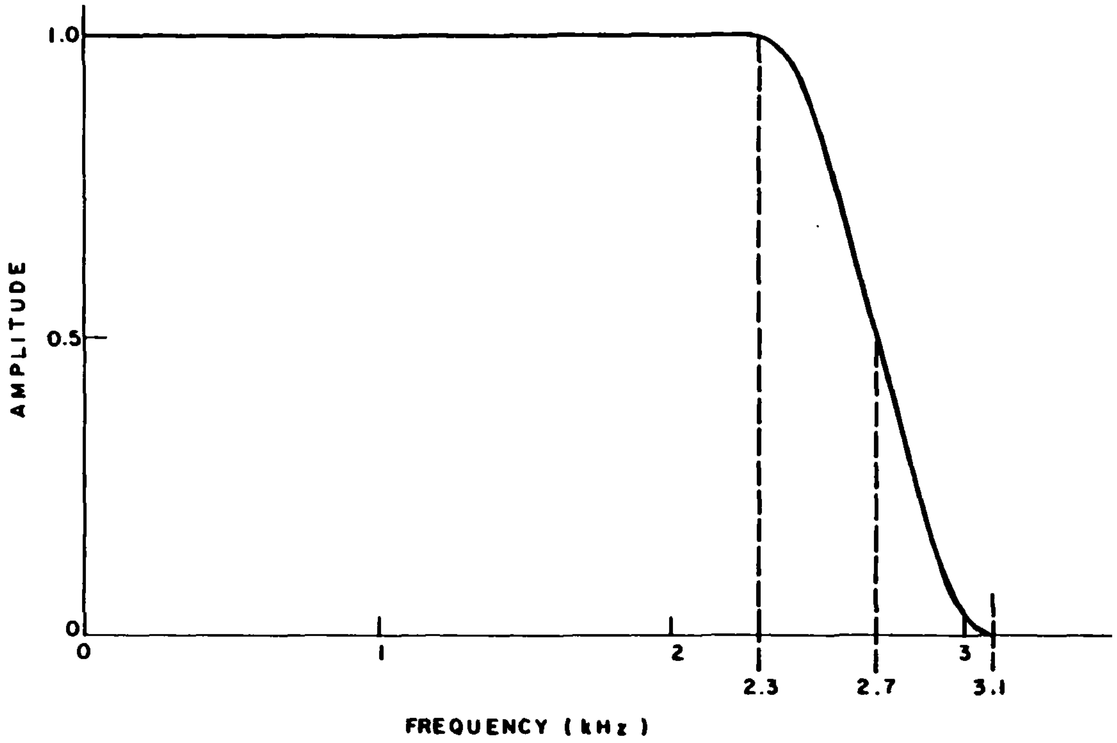


Figure 10. RAISED COSING INTERPOLATING FILTER

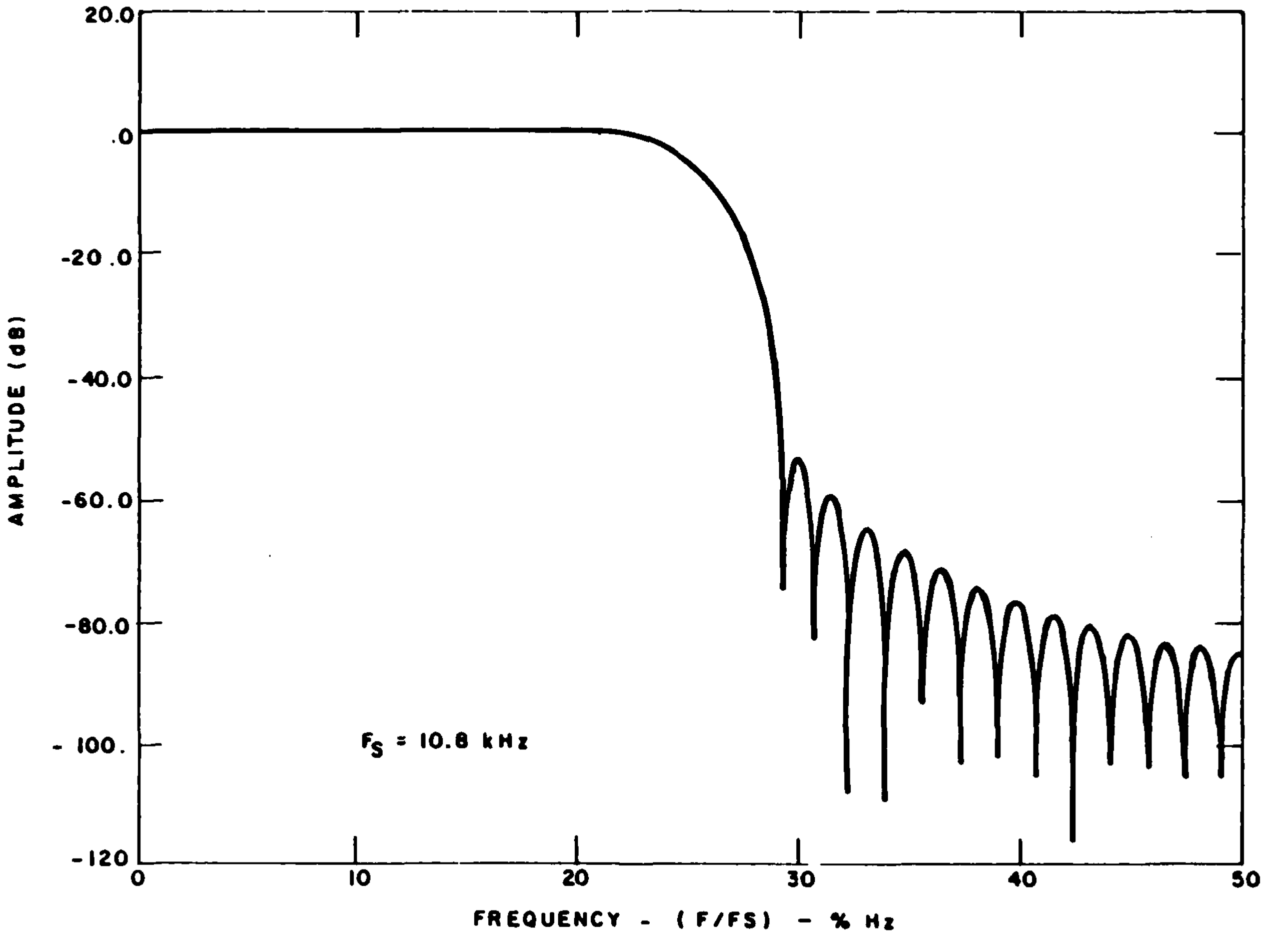


Figure 11. FREQUENCY RESPONSE OF THE INTERPOLATING FILTER

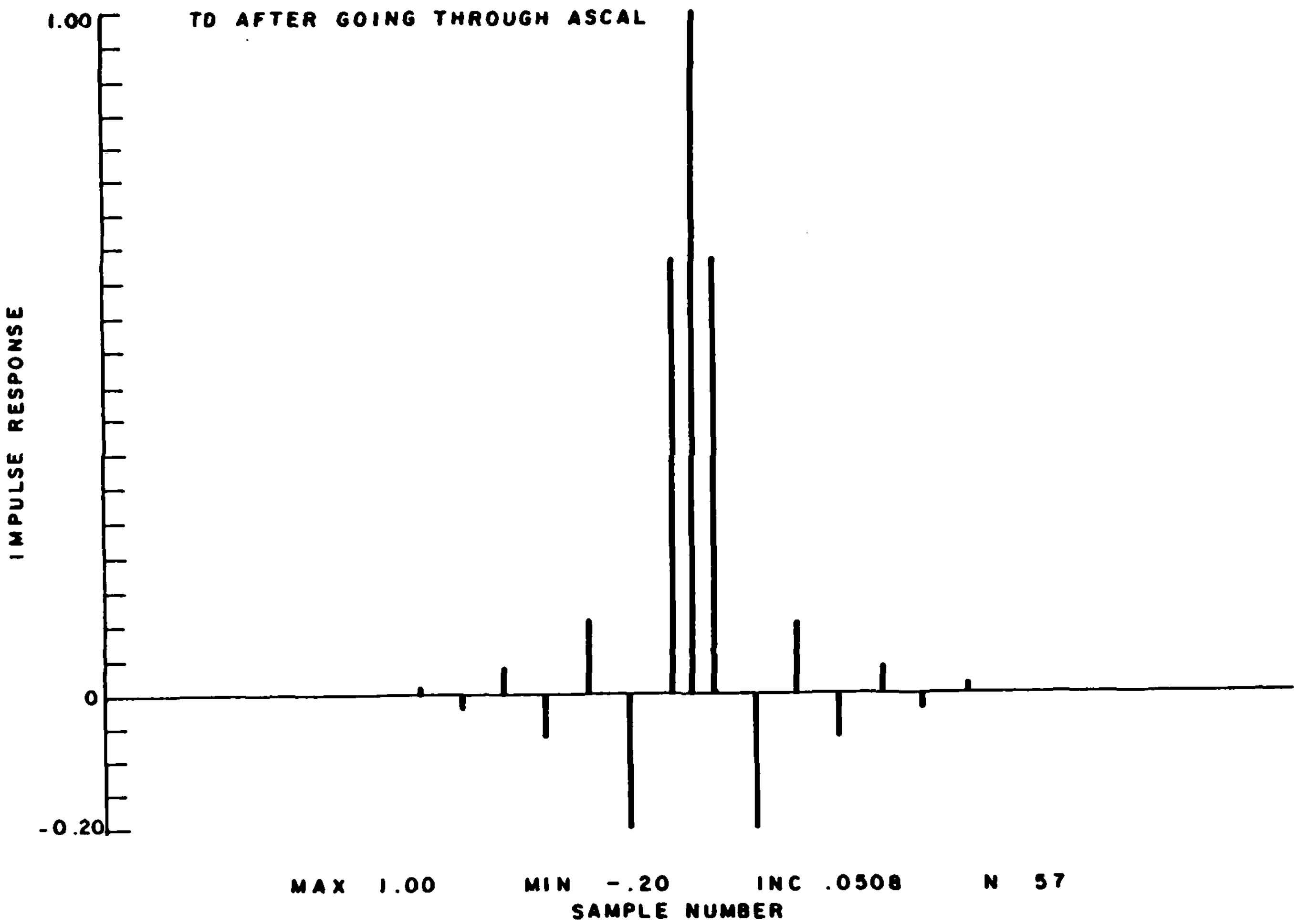


Figure 12. IMPULSE RESPONSE OF THE INTERPOLATING FILTER

TABLE 2
TAP WEIGHTS FOR THE INTERPOLATING FILTER

Tap	Weight (Decimal)	Tap	Weight (Decimal)
1	0	30	20771
2	15	31	0
3	0	32	-6516
4	-38	33	0
5	0	34	3473
6	60	35	0
7	0	36	-2060
8	-69	37	0
9	0	38	1229
10	53	39	0
11	0	40	-696
12	6	41	0
13	0	42	349
14	-131	43	0
15	0	44	-131
16	349	45	0
17	0	46	6
18	-696	47	0
19	0	48	53
20	1229	49	0
21	0	50	-69
22	-2060	51	0
23	0	52	60
24	3473	53	0
25	0	54	-38
26	-6516	55	0
27	0	56	15
28	20771	57	0
29	32767		

and a set of 16-bit binary numbers representing the cosine wave with a frequency of 3000 Hz, also occurring at a 10.8 kHz rate (Figure 8). At this rate the angle between inputs to the mixer increases by the constant

$$\begin{aligned}\theta &= 2\pi f T_s \text{rad} = f T_s \times 360^\circ \\ &= 3 \text{ kHZ} \times \frac{1}{10.8 \text{ kHz}} \times 360^\circ \\ &= 100^\circ\end{aligned}$$

Therefore, the values of the cosine form a sequence which repeats itself after 18 terms (18 x 100° ≈ 1800° = 5 x 360°). These 18 values are stored in a ROM, which is addressed with that sequence of addresses that results in the desired cosine sequence at the mixer input. In the experiment the same equipment was time-shared for use as both the transmitter and receiver mixers.

A phase shift between the transmitter and receiver mixers can be introduced in increments of 20° to measure the effects of the lack of synchronization between mixers.

2.3.3.3 Digital-to-Analog Converter and Analog Filter. The twelve most significant bits of the 16-bit output of the mixer are fed to the DAC, which converts the numbers to an analog voltage waveform that has a frequency spectrum that is essentially nonzero only within the frequency range 300 Hz to 3000 Hz. After a suitable amplification, the signal is ready to be passed on to the telephone line.

2.3.4 Receiver Channel

The receiver channel consists of those circuits that convert the analog waveform (with a 300 to 3000 Hz frequency spectrum) to a 5.4 kHz baseband digital sequence (Figure 8) which is fed either to the equalizer or to the subtractor, depending on whether the equalizer or replicating filter is being used for compensation (Figure 7).

The telephone line simulator attenuates the signal a minimum of 16 dB (Figure 9). A variable gain amplifier was therefore placed after the simulator to enable the voltage level to be raised without exceeding the dynamic range of the ADC in the receiver channel.

The low-pass filter output is fed into the sample-and-hold circuit of a twelve-bit ADC, which converts the analog waveform into a 10.8 kHz digital sequence. The output of the ADC is multiplied by a cosine function having a frequency of 3.0 kHz, using the same mixer employed in the transmitter channel.

The 10.8 kHz mixer output is digitally low-pass filtered to remove unwanted sidebands, then decimated to a 5.4 kHz sequence by deleting every other term in the sequence. The 10.8 kHz output of the first receiver mixer is put through a digital low-pass filter to remove the unwanted sidebands, which, if retained, would produce aliasing in the decimating filter. The filter consists of a transversal filter with 39 taps, each weight represented by 16 bits. It was designed to have its transition band between 2700 Hz and 3300 Hz, a pass band peak-to-peak ripple of less than 1 dB, and a minimum stop band attenuation of 60 dB above the minimum pass band attenuation. The frequency response of this digital low-pass filter is shown in Figure 13. In Figure 14, the amplitude (vertical) axis is expanded to show the pass band ripple of the filter. The filter impulse response is shown in Figure 15. The filter tap weights are listed in Table 3, using octal notation.

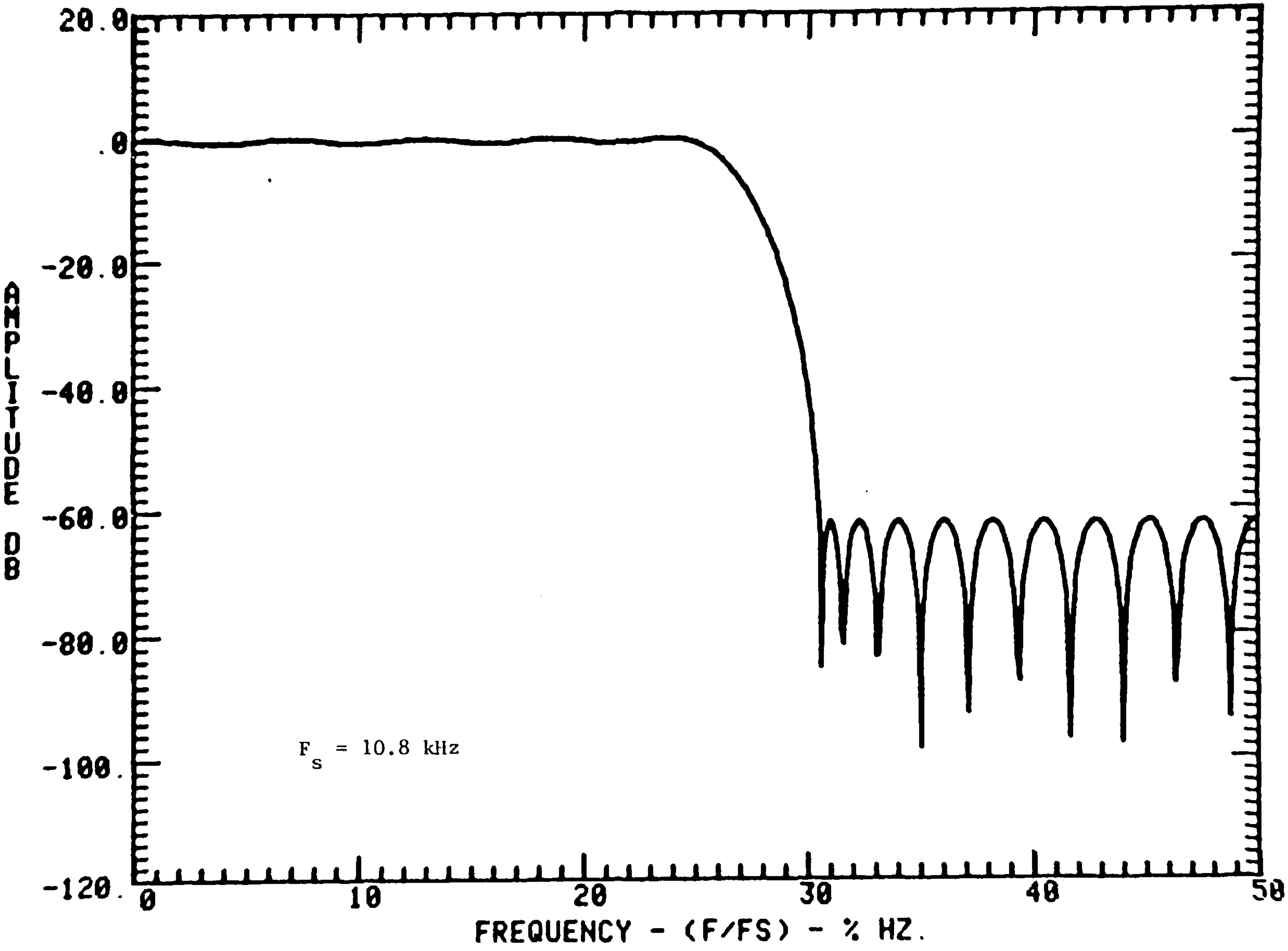


Figure 13. FREQUENCY RESPONSE OF THE DIGITAL LOW PASS FILTER PRECEDING THE DECIMATING FILTER

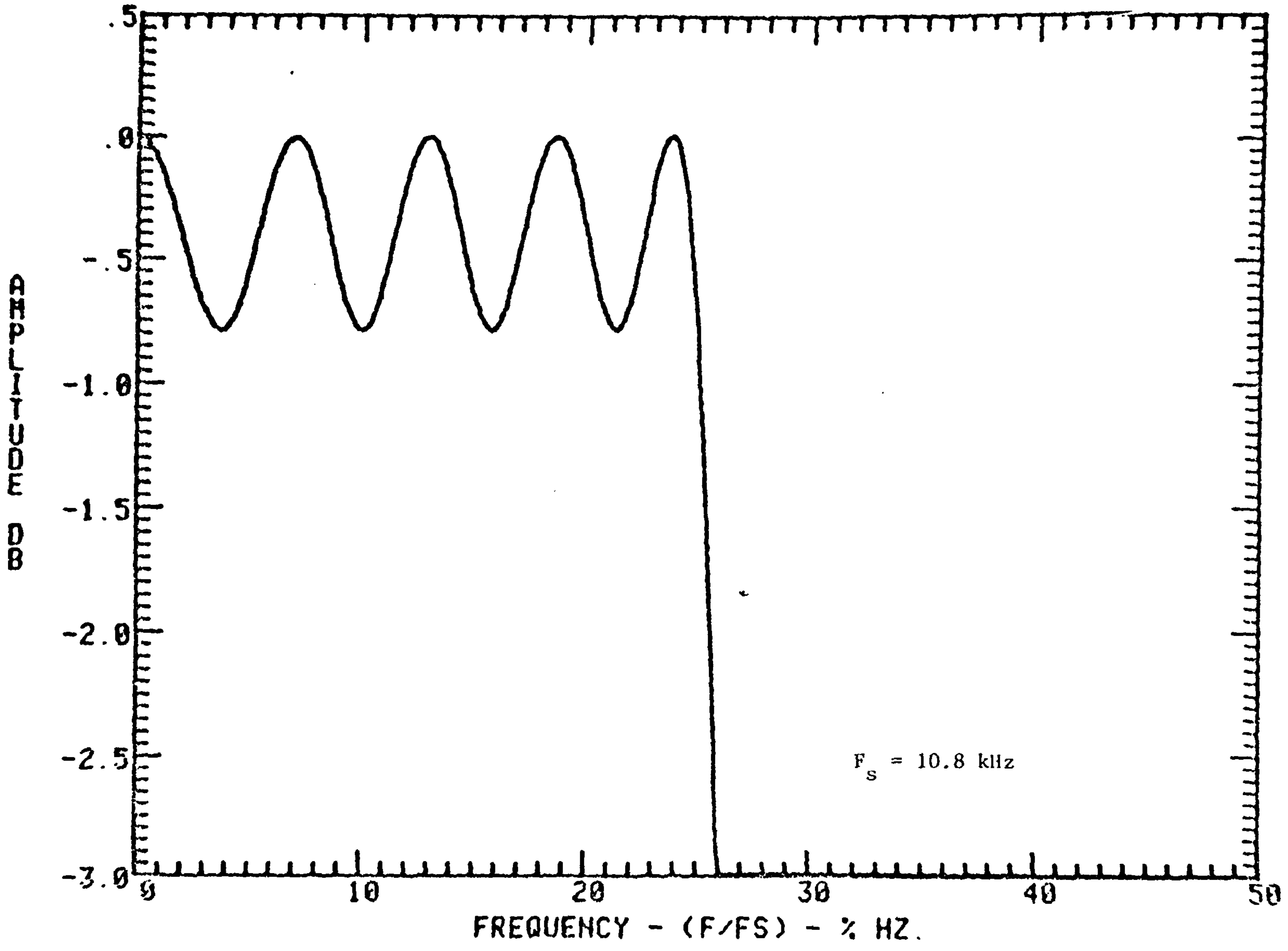


Figure 14. FREQUENCY RESPONSE OF THE LOW PASS FILTER WITH AN EXPANDED AMPLITUDE SCALE TO SHOW THE PASS BAND RIPPLE

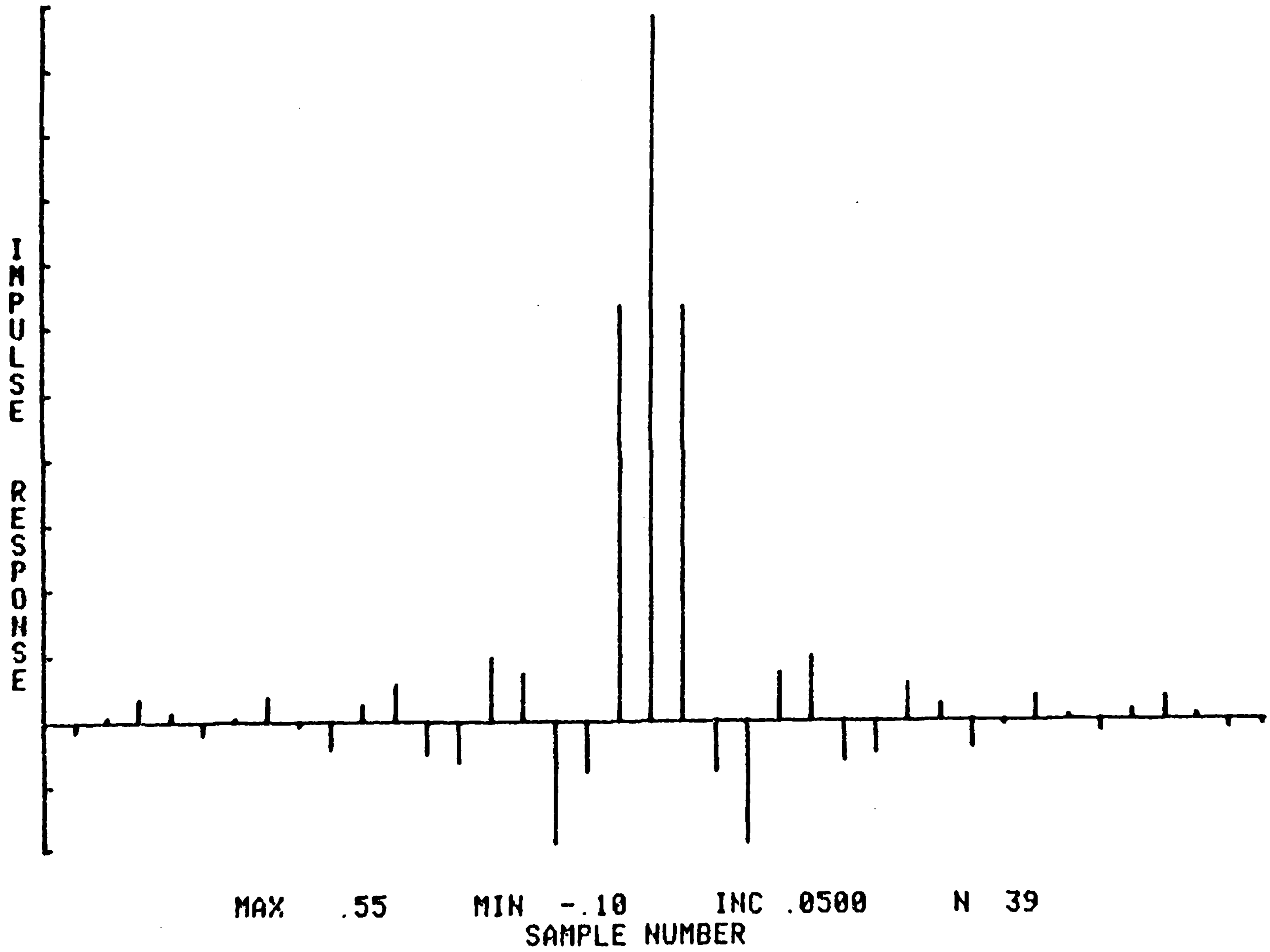


Figure 15. IMPULSE RESPONSE OF THE LOW PASS FILTER PRECEDING THE DECIMATOR

TABLE 3

TAP WEIGHTS OF THE DECIMATOR LOW PASS FILTER

	TAP WEIGHT	
TAP NUMBER	OCTAL NUMBERS	DECIMAL NUMBERS
1	177364	-268
2	177125	-427
3	000362	242
4	002027	1047
5	000724	468
6	176725	-555
7	000223	147
8	002142	1122
9	177456	-210
10	175402	-1278
11	001422	786
12	003210	1672
13	175016	-1522
14	174157	-1937
15	005545	2917
16	004231	2201
17	164475	-5827
18	173316	-2354
19	045253	19115
20	077777	32767
21	045253	19115
22	173316	-2354
23	164475	-5827
24	004231	2201
25	005545	2917
26	174157	-1937
27	175016	-1522
28	003210	1672
29	001422	786
30	175402	-1278
31	177456	-210
32	002142	1122
33	000223	147
34	176725	-555
35	000724	468
36	002027	1047
37	000362	242
38	177125	-427
39	177364	-268

2.4 EXPERIMENTAL PROCEDURES

To prepare the equipment for making measurements, two preparatory steps were taken: set the gains throughout the system to take full advantage of the dynamic ranges of the circuits, and determine the optimum phase shift between the transmitter mixer and receiver mixer.

The tap weights of the replicating filter were then set to cancel the response of an impulse transmitted through the system. With a PN sequence transmitted through the system, the extent to which the replicating filter cancelled the sequence was measured for various numbers of taps.

Using the impulse response of the channel, tap weights for the equalizer were calculated and set into the equipment. Then, with a PN sequence transmitted through the channel, the rms value of the residual PN sequence out of the equalizer was measured.

2.4.1 Initial Settings

2.4.1.1 Dymanic Range Variation. In order to take full advantage of the dynamic ranges available in the individual components of the system, the means were provided to vary the signal strengths at various locations. The points at which the signal strength can be changed are indicated in the block diagram of Figure 16, which shows the locations of the gain controls. In parenthesis are the actual values of the gains (multiplication factors in octal notation) used in the experiment. These values were set by transmitting a PN sequence through the channel and adjusting the gains until the maximum values of the signal at each component was close to the maximum value of which the component is capable of representing.

2.4.1.2 Optimum Phase Shift Between Mixers. When the single-channel configuration was proposed as a simplification over an in-phase and quadrature channel configuration there was some question as to

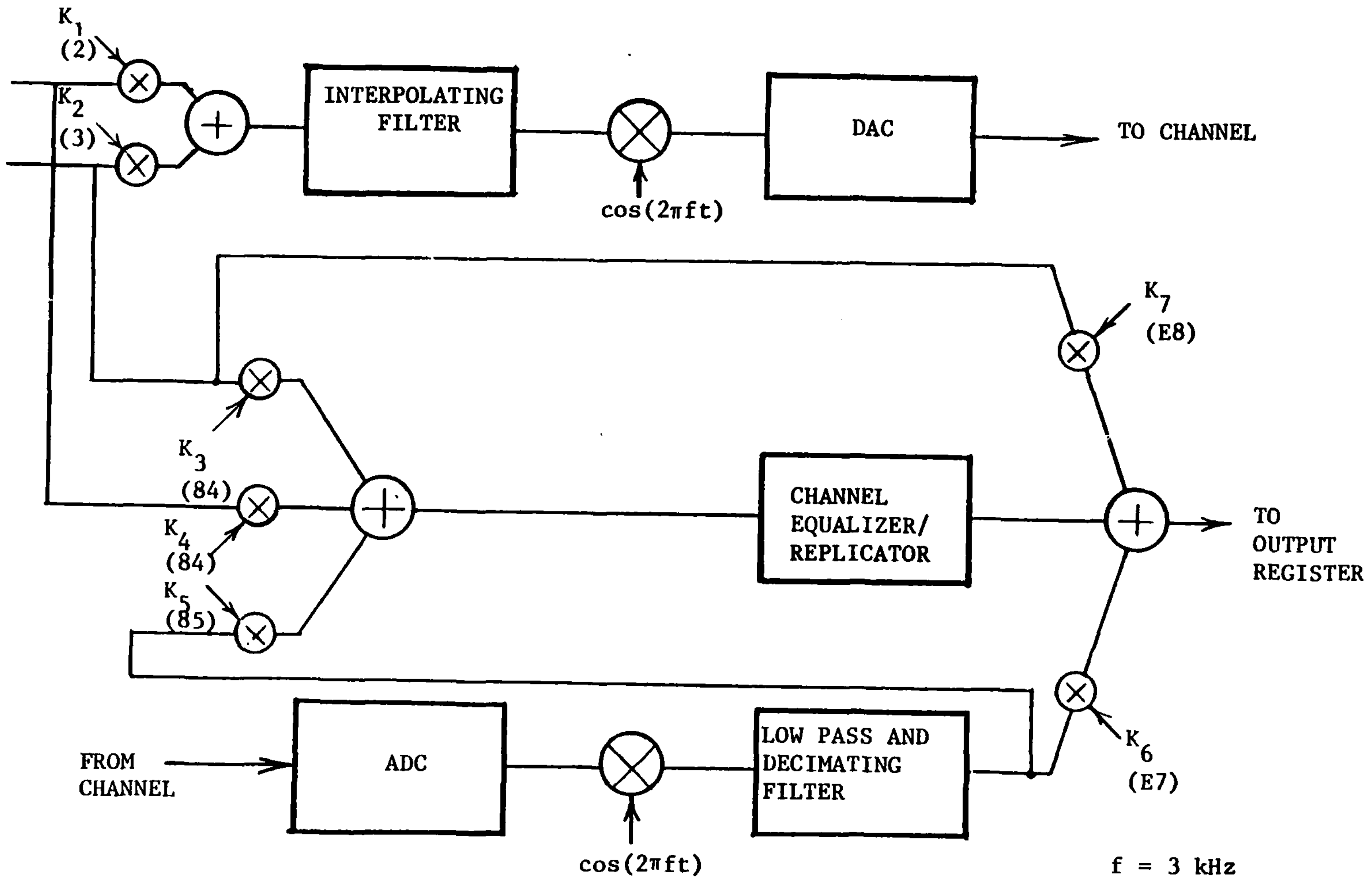


Figure 16. GAIN CONTROL LOCATIONS

whether a single-channel system was a viable concept. The experimental system is an embodiment of the single-channel concept, and expectations of its behavior based on the behavior of single-and double-sideband modulation were confirmed by measurements made on the system.

Because the channel filter does not present infinite attenuation to frequencies above 3 kHz, there is an overlapping of sidebands from the receiver mixer at the lower frequencies. Depending on the relative phase angle between the transmitter and the receiver mixers, the two sideband components will reinforce each other or interfere with each other, resulting in a greater or lesser magnitude of the signal spectrum out of the receiver mixer around 0 Hz.

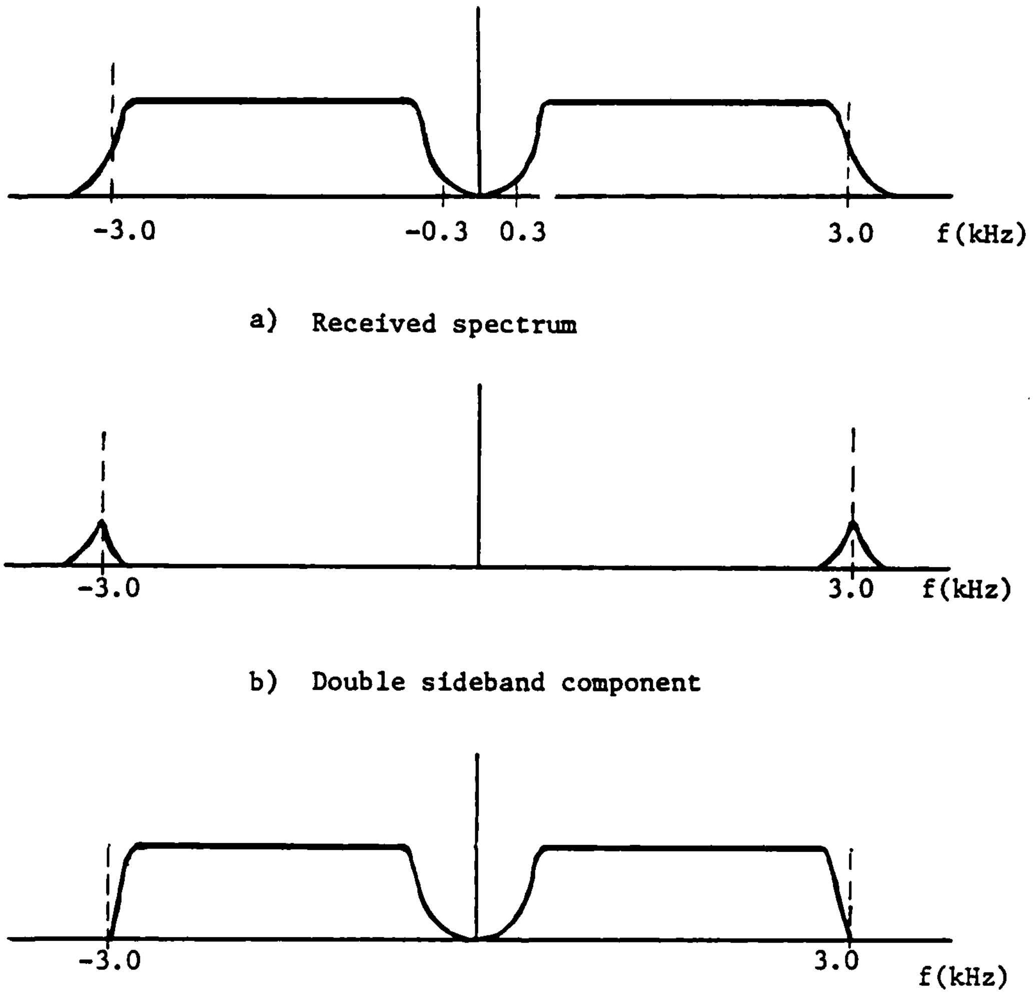
This phenomenon may be understood as demodulation of a double sideband transmission, as compared with that of a single sideband transmission. Suppose that the spectrum at the receiver mixer input looked like that of Figure 17a. Such a spectrum may be decomposed into a double sideband component and a single sideband component as shown in Figures 17b and 17c. The double sideband consists of that component of the spectrum outside the interval (-3 kHz, +3 kHz) augmented by a component within the interval to make the spectrum of positive frequencies symmetrical about 3.0 kHz and the spectrum of negative frequencies symmetrical about -3.0 kHz (Figure 17b). The remainder of the total spectrum is the signal sideband component (Figure 17c).

Consider a component, $X_1(t) = A \cos (2\pi ft)$ of the single sideband component. Out of the transmitter mixer this component becomes

$$X_2(t) = A \cos (2\pi ft) \cos (2\pi f_o t+\emptyset_1)$$

where $\emptyset_1$ is the mixer phase angle. Then

$$X_2(t) = \frac{A}{2} \cos \left[2\pi(f-f_o)t-\emptyset_1 \right] + \frac{A}{2} \cos \left[2\pi(f+f_o)t+\emptyset_1 \right]$$



a) Received spectrum

b) Double sideband component

c) Single sideband component

Figure 17. DOUBLE AND SINGLE SIDEBAND COMPONENTS OF THE RECEIVED SIGNAL SPECTRUM

The upper sideband is suppressed by the channel, so that at the input to the receiver mixer

$$X_3(t) = \frac{A'}{2} \cos\left[2\pi(f-f_o)t-\emptyset_1\right]$$

At the output of the receiver mixer,

$$X_4(t) = X_3(t) \cos\ (2\pi f_o t+\emptyset_2)$$

$$= \frac{A'}{2}\cos\left[2\pi(f-f_o)t-\emptyset_1\right] \cos\ (2\pi f_o t+\emptyset_2)$$

$$= \frac{A'}{4}\cos\left[2\pi(2f_o-f)t+\emptyset_1+\emptyset_2\right] + \frac{A'}{4}\cos(2\pi ft+\emptyset_2-\emptyset_1)$$

The low-pass filter following the mixer suppresses the upper sideband leaving

$$X_5(t) = \frac{A'}{4} \cos\left[2\pi ft+(\emptyset_2-\emptyset_1)\right]$$

The effect of a phase difference, $(\emptyset_2-\emptyset_1)$, in the mixers is only to shift the phase of the single sideband component.

For the double sideband case, consider components, $X_3(t)$, of the signal out of the channel, offset from f_o by a small amount, f, such that

$$X_3(t) = \frac{A'}{2} \cos\left[2\pi(f_o+f)t + \emptyset_1\right] + \frac{A'}{2} \cos\left[2\pi(f_o-f)t + \emptyset_1\right]$$

which differs from $X_3(t)$ in the single sideband case only by the addition of the upper sideband term (the first term).

$$X_4(t) = X_3(t)\ \cos\ (2\pi f_o t+\emptyset_2)$$

$$= A'\ \cos(2\pi f_o t+\emptyset_1)\cos\ (2\pi ft)\ \cos\ (2\pi f_o t+\emptyset_2)$$

$$= \frac{A'}{2}\ \cos\ (2\pi ft)\ \left[\cos\ (\emptyset_2-\emptyset_1) + \cos\ (4\pi f_o t+\emptyset_1+\emptyset_2)\right]$$

The low-pass filter suppresses the double frequency term, so that

$$X_5(t) = \frac{A'}{2}\cos\ (2\pi ft)\ \cos\ (\emptyset_2-\emptyset_1)$$

The effect of a phase difference in the mixers for the double sideband components is greater than for the single sideband components. If the phase difference is 0^o, the amplitude is twice that of the single sideband component. If the phase difference is 90^o, the double sideband component vanishes.

Phase effects in the channel have been neglected in this simple analysis, in order to stress the different effects of the mixer phase difference on the single and double sidehand components.

The mixer phase difference can be varied in steps of 20° with the equipment. The frequency response of the channel was determined for various phase differences by measuring the amplitude of sine waves put through the channel. It was found that for a phase difference of 600° between the mixers, the frequency response around 0 Hz was the highest, while for 700° it was the lowest. The measured frequency response for these two phase differences are plotted in Figure 18, along with the magnitude of the frequency response of the simulated telephone line alone without the mixers or low-pass filter following the receiver mixer. However, the response of the telephone line simulator in the figure is shifted 3 kHz and

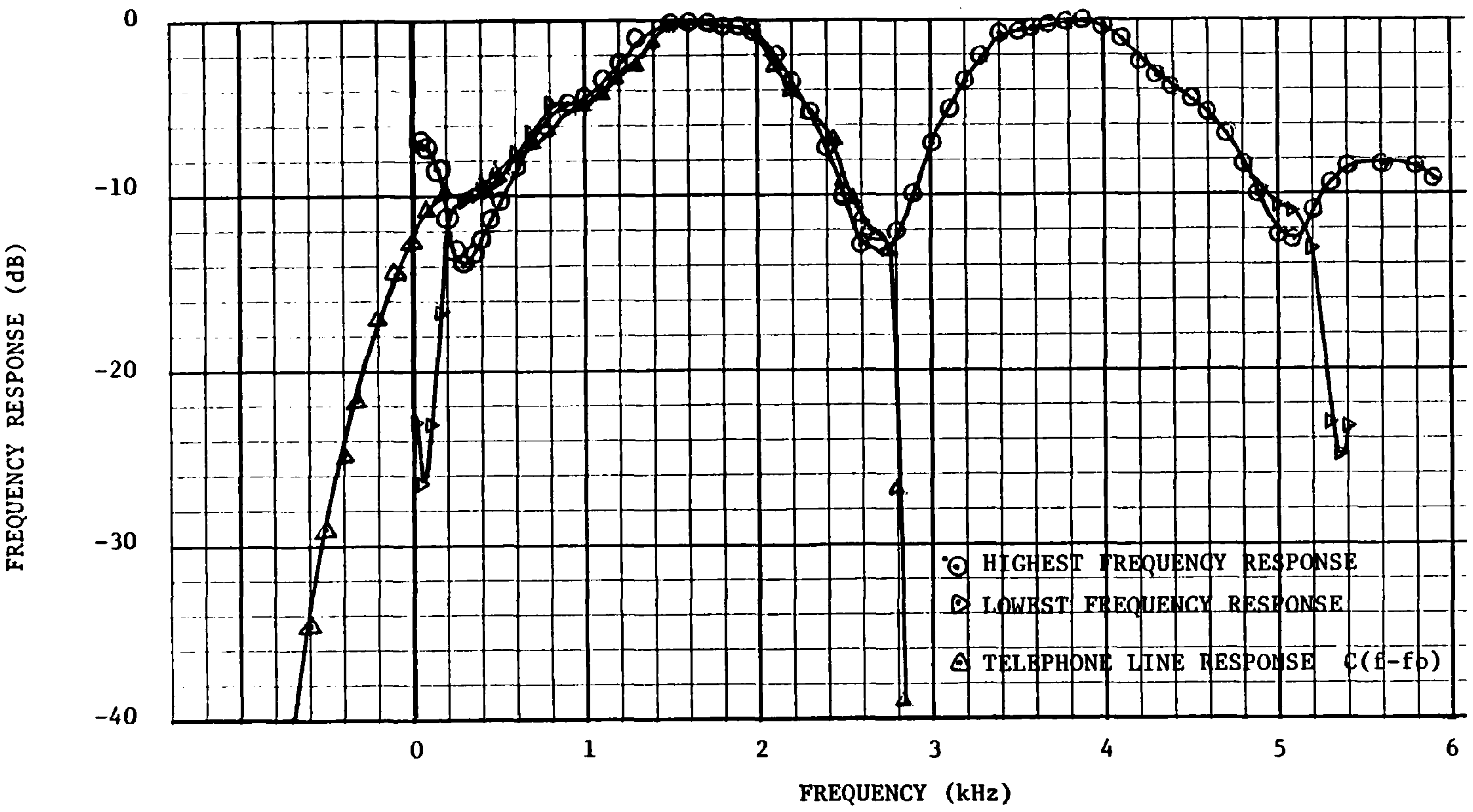


Figure 18. HIGHEST AND LOWEST FREQUENCY RESPONSES OF THE CHANNEL COMPARED TO THE RESPONSE OF THE TELEPHONE LINE

folded about 3 kHz, so that instead of the magnitude of the frequency response, $|C(f)|$,

$$|C(f_o - f)| = |C^*(f_o + f)| = |C(f + f_o)|$$

is plotted in order to directly compare the total channel response with the mixers included to the response of the telephone line alone. As predicted, the response at 0 Hz for the total channel, -6.4 dB, is twice (6 dB above) that measured using the telephone line alone: C(3 kHz) = -12.4 dB.

For the lowest response, the calculated zero response was not attained at 0 Hz. The dip at 50 Hz suggests that had the phase been shifted by finer increments, greater attenuation at 0 Hz would have been obtained.

The mixer phase difference for the experiments was set at 600°, which maximizes the frequency response of the input signal at the band edges. This setting is not critical, but the chosen setting retains more of the response at the lower edge of the band.

2.4.2 Experimental Measurements

2.4.2.1 Replicating Filter. The hardware was switched to the replicating filter configuration to measure the extent to which this configuration could cancel a PN sequence distorted by the channel.

The optimal tap weights for the replicating filter of length L were found experimentally in a simple manner: each tap weight contributed to the impulse response of the filter in only one clock period. With impulses (separated by more than L clock periods) transmitted through the replicator, each tap was set to cancel one term of the digitized channel response exactly, until L consecutive terms were cancelled.

The ideal replicating filter would cancel the impulse response exactly. In practice, the channel response changes slightly during the time required to set the tap weights (they are set manually in this equipment), resulting in a one-bit residue in the cancelled response for many of the clock periods (see Figure 19).

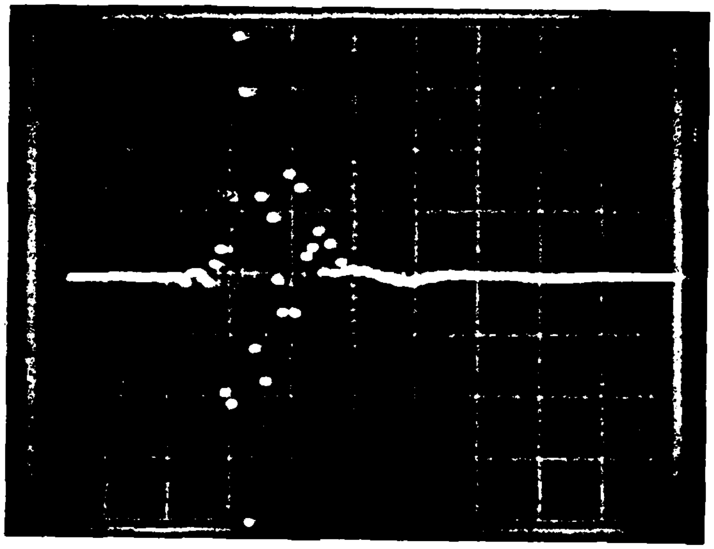
When the timing of the impulse input was varied with respect to the transmitter mixer phase, the impulse response was no longer cancelled exactly at the subtractor in the receiver. The peak of the residual sequence varied with timing from 5 mV to 30 mV, or at most -48 dB from the peak impulse. The cancelled impulse response with the maximum residual is shown in Figure 20.

After the weights were set to give the cancellation (shown in Figure 21), a PN sequence was entered into the channel. The output before the cancellation is shown in Figure 22 and after cancellation in Figure 23. The amount of PN sequence suppression (peak-to-peak) achieved was

$$\frac{6\ \text{V}}{60\ \text{MV}} = 40\ \text{dB}.$$

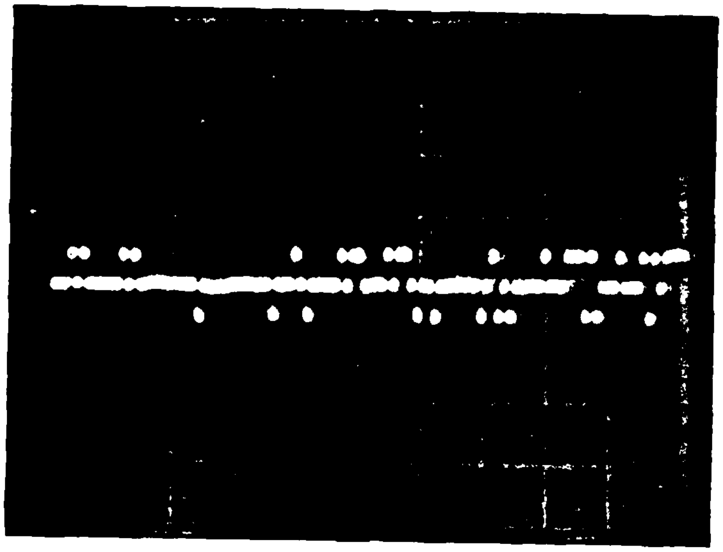
The rms voltage outputs were measured with an rms voltmeter. The suppression of the rms voltage varied with time between 43 dB and 46 dB. The impulse response lasts 73 sampling periods i.e., outside an interval of 73 sampling periods the magnitude of the response is less than one-half of the quantization level. But over the last twelve nonzero values, the magnitude of the impulse response is very low (see Figure 22). When 61 taps were used in the replicating filter instead of 73, there was little increase (1 to 2 dB) in the rms voltage out of the subtractor when a PN sequence was put into the channel.

As the number of taps was decreased below 61 taps, the rms voltage of the cancelled PN sequence increased more rapidly. There-



Horizontal Scale : 2 ms/grid division
Vertical Scale : 0.5 V/grid division

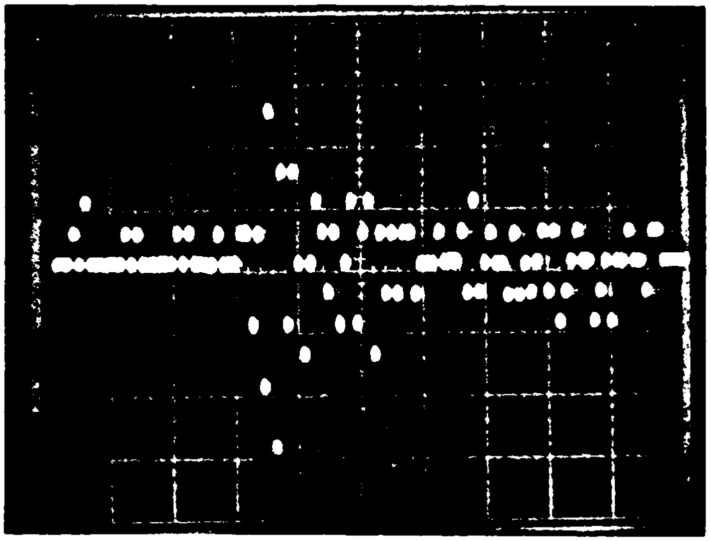
Figure 19. IMPULSE RESPONSE OF THE CHANNEL



Horizontal Scale : 2 ms/grid division

Vertical Scale : 5 mV/grid division

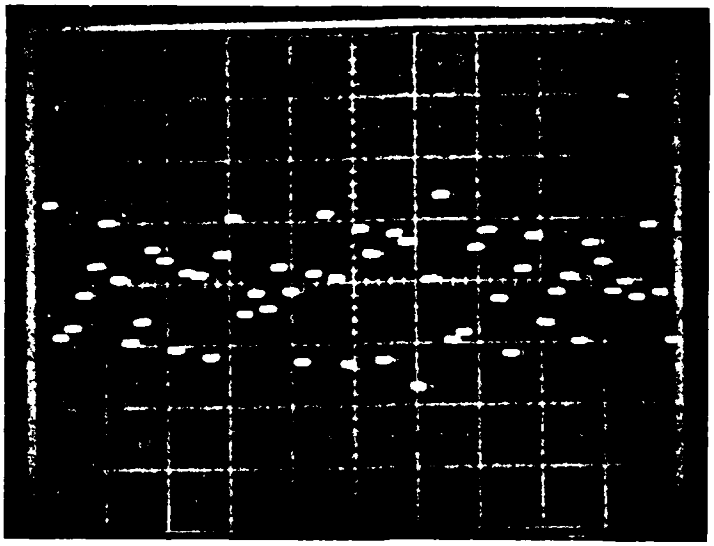
Figure 20. RESIDUE OF THE CANCELLED INPULSE RESPONSE



Horizontal Scale : 2 ms/grid division

Vertical Scale : 5 mV/grid division

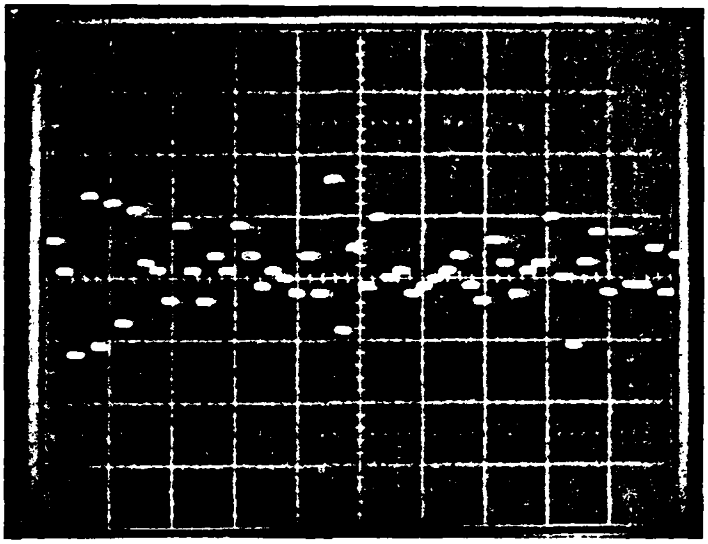
Figure 21. CANCELLED IMPULSE RESPONSE FOR THE WORST-CASE TIMING OF THE IMPULSE WITH RESPECT TO THE TRANSMITTER MIXER PHASE



Horizontal Scale : 1 ms/grid division

Vertical Scale : 2 V/grid division

Figure 22. OUTPUT OF THE SUBTRACTOR WITH A PN SEQUENCE AND NO CANCELLATION



Horizontal Scale : 1 ms/grid division

Vertical Scale : 20 mV/grid division

Figure 23. CANCELLED OUTPUT OF THE SUBTRACTOR WITH A PN SEQUENCE INPUT

fore, 61 taps was considered to be a good compromise between the complexity required and the amount of suppression obtained. Measurements of the cancelled rms voltage as a function of the number of taps in the replicating filter are given in Table 4. Some measurements varied about one percent when the system clock was stopped and restarted. The numbers in the table are estimated averages over several clock restarts in those cases. The uncancelled rms voltage was 2.30 V.

The measurements are plotted in Figure 24 as a function of the number of taps. This figure shows the leveling off of the cancelled voltage as the number of taps is increased.

2.4.2.2 Equalizer. For thc channel replicator, a transversal filter with 61 taps appeared to be a good compromise between complexity and performance. For the sake of comparing the performance of the replicator with that of an equalizer of equivalent complexity, the equalizer was also implemented with a 61-tap transversal filter.

Setting the optimal weights for the equalizer was more complicated than setting the replicator weights because each tap contributed to the equalizer output in many time intervals,and in each time interval the output of the equalizer had contributions from more than one tap. This happens because the input to the equalizer was not an impulse but the impulse response of the channel. In fact, a computer program was written and used to calculate the tap weights.

The channel equalizer (inverse filter) is a filter having a frequency response, H(f), that is ideally the multiplicative inverse of the channel frequency response C(f):

$$H(f) = \left[C(f)\right]^{-1} .$$

The combined frequency response of the two in tandem is, ideally

TABLE 4

CANCELLED RMS VOLTAGE AS A FUNCTION OF THE NUMBER OF TAPS IN THE REPLICATING FILTER

NUMBER OF TAPS	RMS VOLTAGE (MILLIVOLTS)
30	186.5
31	102.2
32	101.4
33	74.9
34	70.3
35	65.1
36	53.2
37	49.7
38	41.3
39	36.6
40	33.2
41	31.3
42	29.3
43	29.3
44	27.7
45	27.1
46	25.9
47	34.3
48	23.6
49	22.4
50	21.4
51	20.6
52	20.4
53	20.3
54	19.6
55	19.6
56	18.65
57	18.85
58	18.35
59	18.59
60	18.23
61	18.43

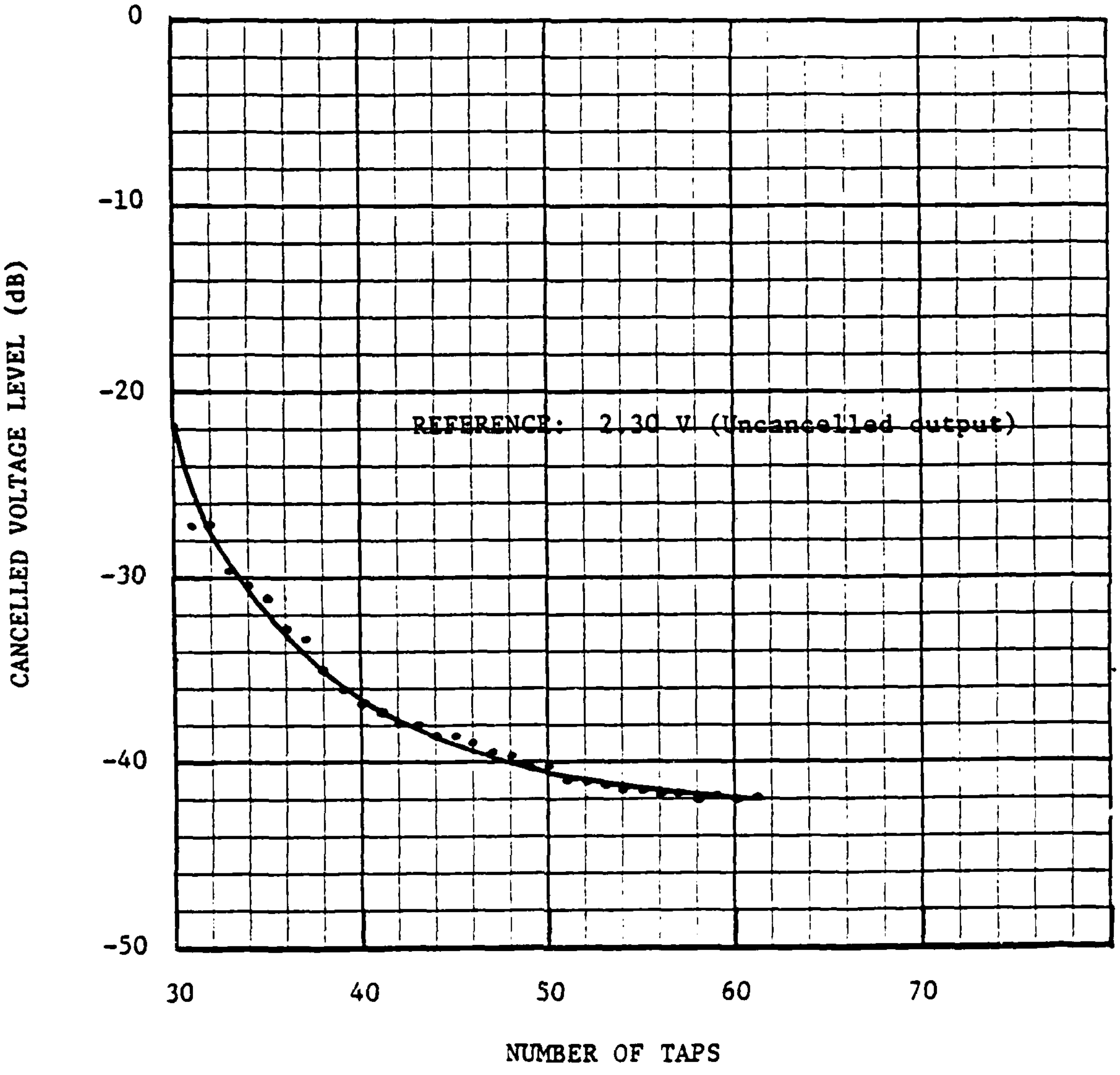


Figure 24. RMS VOLTAGE LEVEL OF THE CANCELLED PN SEQUENCE WITH RESPECT TO THE UNCANCELLED SEQUENCE VS THE NUMBER OF TAPS ON THE REPLICATING FILTER

$$C(f)H(f) = C(f) \left[C(f)\right]^{-1} = 1$$

so that the waveform out of the equalizer is the same as that entering the channel. A computer program was developed to calculate the tap weights for an equalizer from a knowledge of the channel impulse response. The program: (1) calculates the channel frequency response by taking the DFT of the channel impulse response, (2) determines the multiplicative inverse of each value of the channel frequency response to obtain the equalizer frequency response, (3) takes the inverse DFT of the equalizer frequency response to obtain the equalizer impulse response (tap weights), and (4) convolves the impulse response of the channel (input to the equalizer) with the equalizer impulse response to obtain the combined impulse response of the channel and designed equalizer.

A second computer program convolves a PN sequence with the combined impulse response to simulate the restored PN sequence and calculates the rms value of the restored sequence and of the residue serence after subtraction.

A computer-simulated equalizer, using a 61-tap transversal filter, was found by calculation to be capable of suppressing a PN sequence by 39.5 dB. The computer carried through its calculations with 24-bit numbers, compared with the 16-bit numbers employed in the experimental equipment. The 24 bits were suffic..ent to produce negligible round-off error. (In the instances checked, the 24-bit round-off produced no errors in the five most significant bits. It was not determined what the effect on the suppression is of 16-bit rounding). The effects of other quantization errors, such as in the A/D Converter, were also not included.

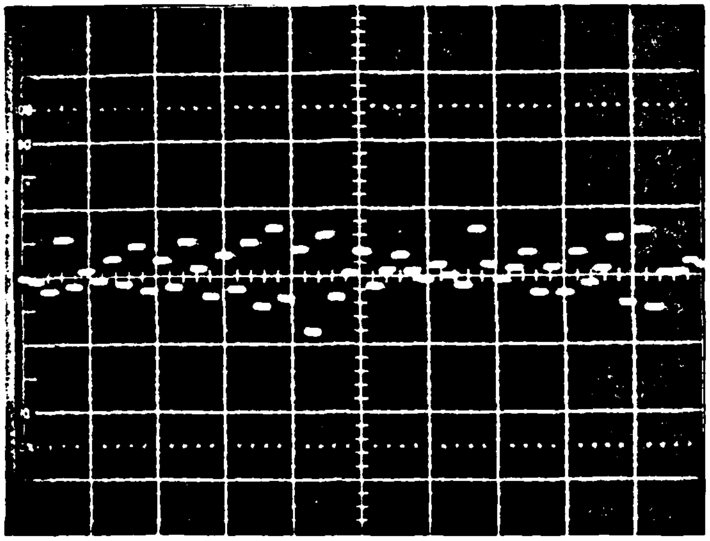
The values of the calculated tap weights were then set in the transversal filter with the equipment in the equalizer configuration.

The lowest measured residual sequence out of the subtractor, after fine tuning the tap weights for minimum residue, was 32.6 dB below the unsuppressed output.

The tap weights were determined by means of a 61-point DFT, which assumes that the input is periodic of period 61. The ideal output of the equalizer would be a sequence of impulses spaced 61 time periods apart. The time sidelobes in the response of the system to each impulse cancel those of neighboring impulses. Therefore, if a single impulse is entered into the channel, the output of the equalizer, designed for a periodic input, includes the uncancelled sidelobes.

There are, however, 61 choices of tap weights which result in the desired impulse train out of the equalizer for a sequence of impulses into the channel. Once a set of tap weights is determined, any circular shift of the tap weights, say a shift of m taps, results in the same impulse train shifted by m time intervals. The rms values of the sidelobes for each of the circular shifts of the tap weights may be compared and that set of tap weights chosen as the optimum set which gives the lowest rms value.

A set of tap weights was calculated as outlined above. This set was then circularly shifted in steps of five taps at a time in both the computer program and the experimental equipment until a minimum residue was reached. The calculated and measured values of the rms residue, compared to the rms value of the uncancelled sequence, are given in Table 5 for a PN sequence entered into the channel. The residue for zero shift is shown in Figure 25. As the tap weights are shifted five taps at a time from 0 to 15 taps both the calculated and measured rms residues decreased. However, the rate of decrease of the measured residue was noticeably less than that of the calculated values (see Table 5). This suggests a limit on performance of the equalizer configuration.



Horizontal Scale : 1 ms/grid division

Vertical Scale : 0.5 V/grid division

Figure 25. RESIDUE OUT OF THE SUBTRACTOR FOR A PN SEQUENCE INPUT AND NO SHIFT OF THE EQUALIZER TAPS

TABLE 5

RESIDUE OF A PN SEQUENCE FOR
CIRCULAR SHIFTS OF THE TAP VALUES

Number of Taps Shifted	Residue*	
	Measured (dB)	Calculated (dB)
0	-25.6	-27.7
5	-27.5	-30.2
10	-29.6	-33.7
15	-32.6	-39.5
20	-30.3	---

* Reference: RMS value of the uncancelled sequence.

The computer programs were run to calculate the suppression possible with an 80-tap equalizer. The suppression was calculated to be 48.3 dB. Table 6 summarizes the results of the calculation and experiment.

TABLE 6

SUPPRESSION OF A PN SEQUENCE OBTAINED
IN THE REPLICATOR AND EQUALIZER CONFIGURATIONS

Number of Taps	Suppression (dB)		
	Replicator	Equalizer	
	(Experimental)	(Experimental)	(Calculated)
61	42	32.6	39.5
73	46	----	----
80	--	----	48.3

2.5 CONCLUSIONS AND RECOMMENDATIONS

When a comparison was made between a 61-tap transversal filter used in a replicator configuration and the same filter used in an equalizer configuration, the former was better in cancelling a PN sequence transmitted through a simulated voice-grade telephone line.

The 42 dB of suppression of the received PN sequence attained with the experimental equipment in the replicator configuration can very likely be increased significantly by three means:

- by using automatic tap weight control which can change the tap weights in less time than it takes for the channel impulse response to drift significantly;

- by identifying the sources of the channel impulse response variation with impulse delay and modifying the system design to remove, or mitigate, the variation, if possible;

- by increasing the number of taps from sixty-one to the same number as the length of the longest expected impulse response (seventy-three for the present simulated channel).

Computer simulations suggest that the equalizer configuration using a 61-tap transversal filter may be capable of up to about 40 dB of suppression of the PN sequence transmitted through the simulated telephone channel, but this was not observed experimentally. This value, however, should be satisfactory for recovering the modulus sequence. Further tests with the experimental equipment will be required to determine the practical limits of the capability of the equalizer configuration.

Four major tasks remain:

1. To add voice to the encryption stream in the transmitter and recover the voice in the receiver with a negligible amount of distortion,

2. to recover the modulus sequence in the receiver at a minimum increase in voice distortion,

3. to implement an algorithm which initially determines the tap weights and updates them periodically, and

4. to test the modem on an actual voice-grade telephone line.

The next phase in the development of the secure voice modem should be the design of a transmitter and receiver employing the replicator configuration for cancelling the PN sequence with voice added, using an adaptive replicator weight control, to attain the first three goals. The equipment should be designed to be readily modified for testing on an actual voice-grade telephone line.

In order to recover the voice with a negligible amount of distortion, the design of the modem should be such as to facilitate the tracing of the origin(s) of the time-varying characteristic of the channel impulse response and to allow modification of the design to lessen the amount of the variance. It is not known how much degradation, if any, is apparent to the ear with the present amount of suppression. But it does not seem worth obtaining more than 54 dB of suppression, since that puts the rms residual encryption at less than one-half of one quantization level.

Two concepts were advanced in [1] for recovering the modulus sequence from the received signal. The "forbidden zone" concept requires decreasing the dynamic range of the voice signal, or, equivalently, decreasing the signal-to-quantization noise ratio; the second concept with its variations allows the voice signal to span the full dynamic range, but may require trading off errors in recovering the modulus sequence with producing distortion in the voice signal. The trade-offs to be made and relative merits of the various alternatives to recover the modulus sequence can be evaluated with the aid of a working modem.

Experience with the present equipment suggests that it may be desirable to implement an adaptive weight control together with the addition of voice-carrying capability in the next step of this development. Setting the weights manually for the purpose of experimental verification has proven to be a tedious, time consuming, and error-prone task, and the slowly time-varying impulse response due to temperature variations requires updating the settings periodically to maintain them at their optimum values.

The final goal is to replace the telephone line simulator with an actual voice-grade telephone line to compare the quality of voice passed through the modem with that of clear voice transmitted over the same line and to find out what modifications of the design are necessary to improve the voice quality.

SECTION III

FINITE FIELD DIGITAL FILTERS

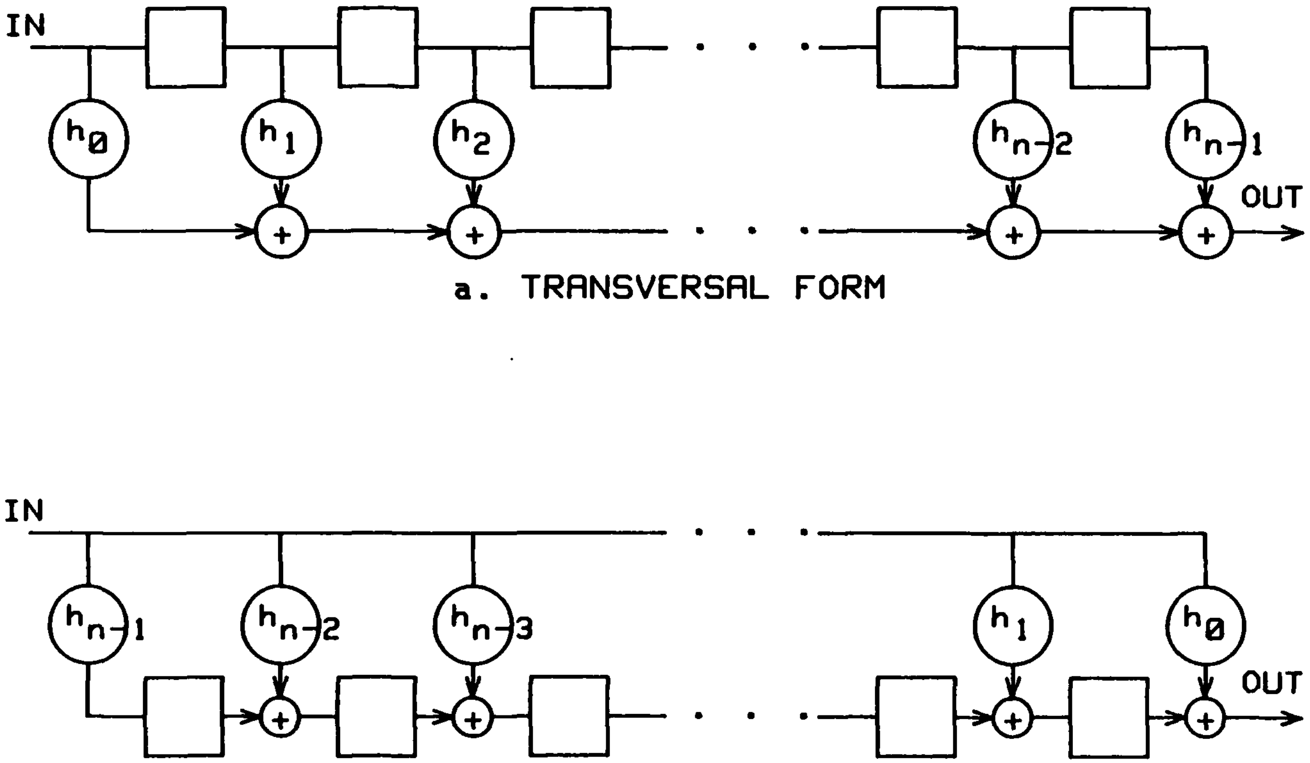
In the previous section, several digital filters were evident in the design of the transmitter and receiver sections of the secure voice modem, and also in the channel compensation techniques discussed. These digital filters are suitable candidates for finite-field design. After a brief discussion of the general nature of finite-field filters, the design of four different filters will be presented.

3.1 GENERAL CONSIDERATIONS

We assume at the outset that the filter to be implemented is specified by its unit sample response in the time domain, given as a sequence of outputs resulting from an impulsive input, and represented conveniently as a power series in a variable D associated with unit delay

$$h(D) = h(0) + h(1)D + h(2)D^2 + \ldots \qquad (3.1)$$

For a finite impulse response (FIR) filter the sequence of impulse response parameters is finite, terminating with some final value $h(n-1)$ for an n-point filter and being zero thereafter. The impulse response parameters $\{h(k)\}$ are assumed to be real numbers that acquire values in the infinite field of rationals. Given the impulse response sequence, an FIR filter may be implemented immediately as a transversal, or tapped delay line, filter as shown schematically in Figure 26a. Alternatively, the filter may be implemented in the transposed form shown in Figure 26b. If the transversal filter has length n, if the input values lie in the range $\{-2^m, 2^m\}$ and if no errors are to occur in the response to arbitrary inputs, then the implementation must be capable of handling numbers up to $2^m \sum_{k=0}^{n-1} |h(k)|$ in magnitude. Ordi-



a. TRANSVERSAL FORM

b. TRANSPOSED FORM

Figure 26. FIR FILTER STRUCTURES

narily, one is faced either with providing sufficient word length in fixed-point arithmetic, resorting to floating-point arithmetic with finite word-length imposing some roundoff error, or accepting the roundoff and truncation error imposed by the complexity of fixed-point implementation that can be afforded.

The finite-field approach to FIR filtering trades off the complexity required in full-precision fixed-point arithmetic, chiefly expended on the multipliers, with a parallel residue processor in which the individual components are simpler; although there are more of them. In this approach, the impulse response sequence of equation (3.1) is first mapped into a finite ring of integers and the ring is decomposed into the direct sum of simpler components, in our case finite fields, a filter being implemented in each of the component fields. If the integer ring is selected to be large enough to contain all of the numerical values, then the component field computations will not introduce any further computational error from roundoff or truncation since the field operations are finite and closed. This is true even in the case of recursive implementation, containing feedback. Ordinarily roundoff and truncation errors would be disastrous in effect in a filter with feedback. Some error inevitably will occur initially in the process of mapping the impulse response into a finite ring, but that error can be controlled in the design process by making the ring modulus large enough for the error to be negligible. Once the initial error is accounted for, there is no further computational error provided that the finite ring is large enough to contain all the numerical values. This last point is important; the numerical values occurring in computation must not overflow the ring.

Assume that the impulse response values of equation (3.1) are elements of a finite ring of integers modulo M, designated Z_M. Let the ring-modulus M be expressed as the product of distinct primes

$$M = \prod_{i=1}^{L} p_i \tag{3.2}$$

M being chosen in this way to contain the numerical values of computation. The Chinese remainder theorem [2] may be used to show that the system of congruences

$$h_i(k) \equiv h(k) \bmod p_i \;;\quad i = 1, 2, \ldots, L \tag{3.3}$$

has a unique solution:

$$h(k) = \sum_{i=1}^{L} h_i(k) M_i M_i^{-1} \tag{3.4}$$

where $M_i = M/p_i$ and

$$R_i = M_i M_i^{-1} \equiv 1 \bmod p_i \tag{3.5}$$

The multiplicative inverse M^{-1} may be calculated by Euclid's algorithm, or equivalently by a recursive calculation based on a continued-fraction approximation. Fermat's theorem also could be used to determine M_i^{-1} by exponentiation of M_i.

The ring of integers modulo M is isomorphic to the external direct product of the finite fields $GF(p_i) = Z_{p_i}$; $i = 1, \ldots, L$. Consequently, linear operations over Z_M can be carried out in the component fields $GF(p_i)$ with the results mapped back into the product ring Z_M by means of the Chinese remainder theorem composition formula. This procedure, known as "residue number processing" is shown schematically in Figure 27 for which six primes have been specified.

For convolution,

$$c(\ell) = \sum_{j=0}^{n-1} u(j)h(\ell-j) \tag{3.6}$$

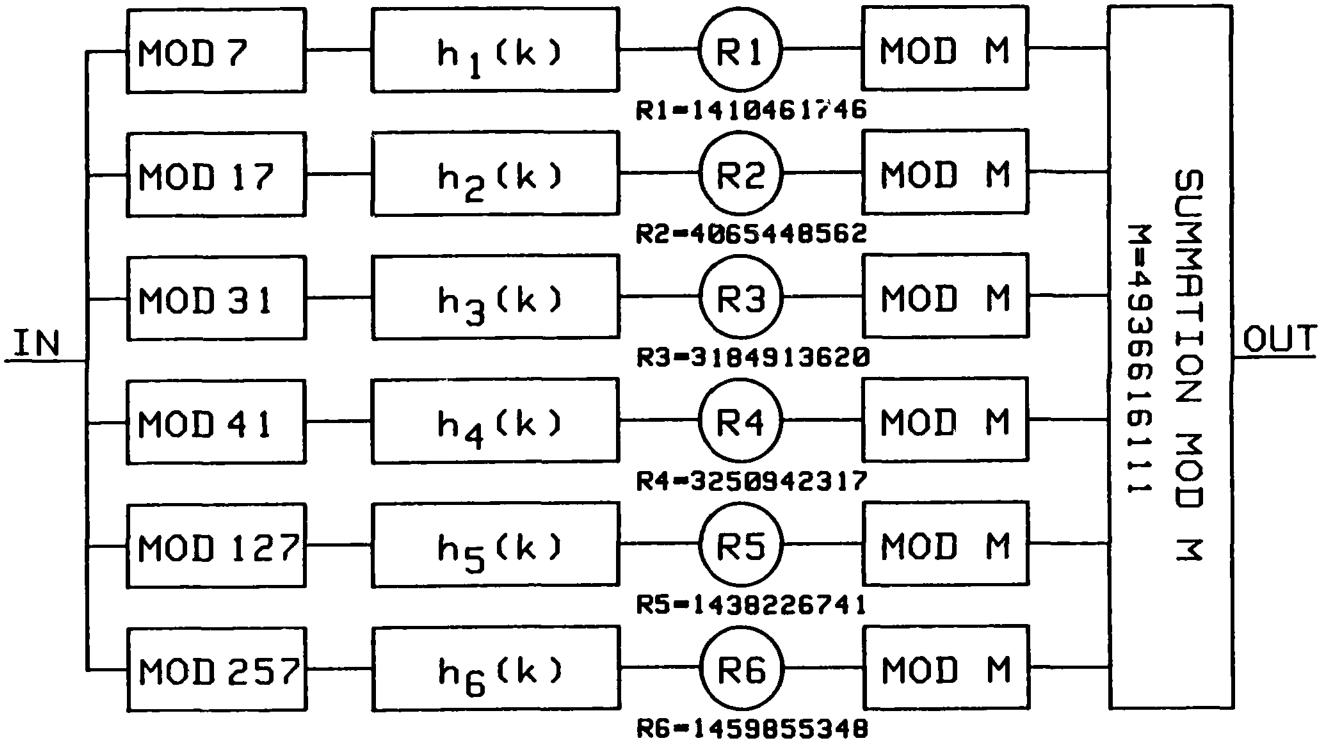


Figure 27. A DIGITAL FILTER DECOMPOSED INTO FINITE FIELD COMPONENTS

in which the input samples $\{u(\cdot)\}$, impulse response samples $\{h(\cdot)\}$ and output samples $\{c(\cdot)\}$ are members of the ring Z_M, the component convolutions

$$c_i(\ell) = \sum_{j=0}^{n-1} u_i(j)b_i(\ell-j)$$

are computed over the finite fields $GF(p_i) = Z_{p_i}$ where, for $i = 1,\ldots,L$

$$\begin{aligned} u(\cdot) &\equiv u_i(\cdot) \bmod p_i \\ h(\cdot) &\equiv h_i(\cdot) \bmod p_i \qquad (3.7) \\ c(\cdot) &\equiv c_i(\cdot) \bmod p_i \end{aligned}$$

By the division algorithm, the congruence relation can also be expressed as

$$u(\cdot) = q_i p_i + u_i(\cdot) \qquad (3.8)$$

and it is assumed that

$$M = \prod_{i=1}^{L} p_i \; ; \; p_i \neq p_j \quad \text{for } j \neq i. \qquad (3.9)$$

We may regard the input, impulse response, and output sequences as polynomials in the delay variable, as in equation (3.1). The operation of convolution is then the same as polynomial multiplication, e.g.,

$$c(D) = u(D)h(D) \qquad (3.10)$$

represents convolution over the integer ring Z_M and

$$c_i(D) = u_i(D)h_j(D) \qquad (3.11)$$

represents convolution over the finite field GF(p_i). Much of the recent work in digital signal processing is concerned with fast algorithms, based on discrete Fourier-like transforms, to compute convolutions by successive transformation employing FFT-like decimation algorithms. The transform method applies to cyclic convolution. In this section we are concerned only with direct computation of linear convolution.

The example illustrated in Figure 27 shows a finite ring whose modulus M = 4936616111 is decomposed into the direct sum of prime fields whose characteristics are respectively 7,17,31,41,127,257. While the original ring would require all computational results to be carried to at least 32 bits, the component sections require finite field arithmetic with the equivalent of 8-bit or lesser precision, greatly simplifying the hardware complexity of the multipliers. In such an arrangement, in order to avoid overflow the filter coefficients and input sample values must be suitably restricted, but once the dynamic range constraints have been established there is no further source of roundoff error, barring hardware faults. The design of two filters presented in Section II will next be described in order to illustrate the design procedure. In both cases we will assume that the finite field digital filter decomposition of Figure 27 will be used.

3.2 LOWPASS FILTER EXAMPLES

Two lowpass filters were used in the experimental design of the secure voice modem discussed in Section II. They are the digital interpolating filter and anti-aliasing filter shown in the block diagram of Figure 8. Each of these filters implements a direct convolution in hardware that used fixed-point arithmetic with a 16-bit multiplier and accumulator. The impulse response coefficients were presented in Table 2 for the interpolating filter and in Table 3 for the anti-aliasing filter. In both cases the input sample values were restricted by an attenuation control to avoid objectionable roundoff effects. The finite field design of these filters is discussed below.

3.2.1 Digital Interpolating Filter

The impulse response (or unit sample response) corresponding to the values in Table 2 was plotted in Figure 12. The corresponding frequency response was shown in Figure 13. The filter was designed as a windowed sinc function sampled at the Nyquist rate using n = 57 coefficients of which only the middle and even-numbered coefficients are non-zero. The zero-valued coefficients correspond to the zero-crossing points of the sinc function. The maximum-value coefficient is 32767 or $2^{15}-1$. Consequently, since all coefficients are integers taking either positive or negative values, they all lie in a finite ring mod 2^{16}. If the filter were implemented in transversal form as shown in Figure 26, then the magnitude of the largest value occurring in the output cannot exceed $2^m \sum_{i=1}^{n} |h_k|$ for (m + 1)-bit quantization of the input samples. Computation, using the values of Table 2, establishes that $\sum_{i=1}^{57} |h(k)| = 103699$. The ring modulus used is $M = 4936616111 \approx 2^{32}$. If the input samples are restricted in magnitude to ~47605, overflow errors cannot occur. Clearly, the finite field filter decomposition of Figure 27 will support 16-bit quantization of the input samples.

The impulse response coefficients in the component fields are tabulated in Table 7. The tabulated coefficients in each field are exactly the tap weights that would be used in the component filters, all arithmetic being carried out in the respective field. It is necessary, of course, to map the input samples to each filter into the same field and finally to reconstruct the results in the integer ring. These steps require arithmetic operations of greater hardware

TABLE 7

UNIT SAMPLE RESPONSE OF THE FINITE FIELD COMPONENTS

OF A DIGITAL INTERPOLATING FILTER

k	h(k)	MOD 7	MOD 17	MOD 31	MOD 41	MOD 127	MOD 257
1	0	0	0	0	0	0	0
2	15	1	15	15	15	15	15
3	0	0	0	0	0	0	0
4	-38	4	13	24	3	89	219
5	0	0	0	0	0	0	0
6	60	4	9	29	19	60	60
7	0	0	0	0	0	0	0
8	-69	1	16	24	13	58	188
9	0	0	0	0	0	0	0
10	53	4	2	22	12	53	53
11	0	0	0	0	0	0	0
12	6	6	6	6	6	6	6
13	0	0	0	0	0	0	0
14	-131	2	5	24	33	123	126
15	0	0	0	0	0	0	0
16	349	6	9	8	21	95	92
17	0	0	0	0	0	0	0
18	-696	4	1	17	1	66	75
19	0	0	0	0	0	0	0
20	1229	4	5	20	40	86	201
21	0	0	0	0	0	0	0
22	-2060	5	14	17	31	99	253
23	0	0	0	0	0	0	0
24	3473	1	5	1	29	44	132
25	0	0	0	0	0	0	0
26	-6516	1	12	25	3	88	166
27	0	0	0	0	0	0	0
28	20771	2	14	1	25	70	211
29	32767	0	8	0	8	1	128
30	20771	2	14	1	25	70	211
31	0	0	0	0	0	0	0
32	-6516	1	12	25	3	88	166
33	0	0	0	0	0	0	0
34	3473	1	5	1	29	44	132
35	0	0	0	0	0	0	0
36	-2060	5	14	17	31	99	253
37	0	0	0	0	0	0	0
38	1229	4	5	20	40	86	201
39	0	0	0	0	0	0	0
40	-696	4	1	17	1	66	75
41	0	0	0	0	0	0	0
42	349	6	9	8	21	95	92
43	0	0	0	0	0	0	0
44	-131	2	5	24	33	123	126
45	0	0	0	0	0	0	0
46	6	6	6	6	6	6	6
47	0	0	0	0	0	0	0
48	53	4	2	22	12	53	53
49	0	0	0	0	0	0	0
50	-69	1	16	24	13	58	188
51	0	0	0	0	0	0	0
52	60	4	9	29	19	60	60
53	0	0	0	0	0	0	0
54	-38	4	13	24	3	89	219
55	0	0	0	0	0	0	0
56	15	1	15	15	15	15	15
57	0	0	0	0	0	0	0

complexity, especially in the reconstruction, but there are relatively few components. Since the filter coefficients are defined as integers and there is no overflow, all convolutions with the quantized input samples are exact; there are no sources of computational error.

3.2.2 Anti-Aliasing Filter

The second example is the anti-aliasing filter of Section II, also referred to as a decimator. The impulse response coefficients shown in Table 3 were plotted in Figure 15 and the corresponding frequency response was shown in Figure 13 and Figure 14. The filter was designed to a prescribed frequency response using the minimax procedure discussed by Parks and McClellan [3].

For this filter, calculations show that $\sum_{i=1}^{39} |h_k| = 120957$. Consequently, overflow conditions are not possible if the input sample magnitude is less than 40812. Again, the configuration of Figure 27 will support 16-bit quantization of the input samples.

The finite field values of the tap weights for the transversal filter implementation in each of the prime fields is given in Table 8. As in the case of the interpolating filter, there is no source of computational error.

3.3 FINITE FIELD CONVOLUTION

In Section II the results of an experiment to cancel a PN sequence transmitted over a simulated time-dispersive channel were described. Two methods of channel compensation were tested, one which equalized the channel by means of an inverse filter, and the other which compensated for time-dispersion by replicating the channel so as to prefilter the reference sequence prior to cancellation. The experiment favored the second method, assuming of course

TABLE 8

UNIT SAMPLE RESPONSE OF THE FINITE FIELD COMPONENTS

OF A DIGITAL ANTI-ALIASING FILTER

k	h(k)	MOD 7	MOD 17	MOD 31	MOD 41	MOD 127	MOD 257
1	-268	5	4	11	19	113	246
2	-427	0	15	7	24	81	87
3	242	4	4	25	37	115	242
4	1047	4	10	24	22	31	19
5	468	6	9	3	17	87	211
6	-555	5	6	3	19	80	216
7	147	0	11	23	24	20	147
8	1122	2	0	6	15	106	94
9	-210	0	11	7	36	44	47
10	-1278	3	14	24	24	119	7
11	786	2	4	11	7	24	15
12	1672	6	6	29	32	21	130
13	-1522	4	8	28	36	2	20
14	-1937	2	1	16	31	95	119
15	2917	5	10	3	6	123	90
16	2201	3	8	0	28	42	145
17	-5827	4	4	1	36	15	84
18	-2354	5	9	2	24	59	216
19	19115	5	7	19	9	65	97
20	32767	0	8	0	8	1	128
21	19115	5	7	19	9	65	97
22	-2354	5	9	2	24	59	216
23	-5827	4	4	1	36	15	84
24	2201	3	8	0	28	42	145
25	2917	5	10	3	6	123	90
26	-1937	2	1	16	31	95	119
27	-1522	4	8	28	36	2	20
28	1672	6	6	29	32	21	130
29	786	2	4	11	7	24	15
30	-1278	3	14	24	34	119	7
31	-210	0	11	7	36	44	47
32	1122	2	0	6	15	106	94
33	147	0	11	23	24	20	147
34	-555	5	6	3	19	80	216
35	468	6	9	3	17	87	211
36	1047	4	10	24	22	31	19
37	242	4	4	25	37	115	242
38	-427	0	15	7	24	81	87
39	-268	5	4	11	19	113	246

that the channel is linear and constant. In either case the performance is limited in addition by adequate estimation or measurement of the channel's impulse response and by computational or roundoff errors that may occur during filtering. In this section we will describe the finite field design of both the channel replication filter and its inverse filter. We will also present the calculated impulse response of the pair in tandem which simulates the impulse response of the equalized channel. The simulated cancellation of a PN sequence after transmission through the compensated channel will be presented. These filtering examples will assume the finite field implementation of Figure 27. Results for PN sequence cancellation will be compared with conventional implementation using fixed-point arithmetic with different levels of precision.

3.3.1 Channel Replication Filter

The impulse response parameters of the channel replicating filter discussed in Section II are tabulated in Table 9. Their decomposition into finite field components is shown in Table 10. The impulse response, placing in evidence the time-dispersive character of the channel, is shown in Figure 28. The corresponding frequency response is shown in Figure 29. In both cases the time and frequency scales are normalized for convenience. It should be noted that the impulse response includes the entire channel, from transmitter channel input to receiver channel output, as shown in the block diagram of Figure 8. It includes, therefore, somewhat more than the telephone line simulator.

As shown in the values of Table 9, the replicating filter's impulse response coefficients are integers that can be represented with 15 or fewer bits. It can be verified that the ring modulus

TABLE 9

UNIT SAMPLE RESPONSE OF THE CHANNEL
REPLICATING FILTER

k	h(k)	k	h(k)
1	16	31	-386
2	16	32	-864
3	-67	33	-464
4	0	34	-464
5	192	35	-416
6	-24	36	-288
7	-208	37	-304
8	36	38	-192
9	56	39	-160
10	-33	40	-96
11	-1392	41	-96
12	1957	42	-16
13	5993	43	-64
14	-4147	44	32
15	-8635	45	-48
16	4912	46	48
17	3568	47	-48
18	-2992	48	48
19	4352	49	-32
20	-336	50	32
21	-1408	51	-48
22	1840	52	0
23	-4176	53	-48
24	1488	54	0
25	-3040	55	-48
26	-720	56	0
27	-1264	57	-48
28	-1616	58	0
29	-464	59	32
30	-1424	60	0
		61	0

TABLE 10

UNIT SAMPLE RESPONSE OF THE FINITE FIELD COMPONENTS OF THE CHANNEL REPLICATING FILTER

k	h(k)	MOD 7	MOD 17	MOD 31	MOD 41	MOD 127	MOD 257
1	16	2	16	16	16	16	16
2	16	2	16	16	16	16	16
3	-67	3	1	26	15	60	190
4	0	0	0	0	0	0	0
5	192	3	5	6	28	65	192
6	-24	4	10	7	17	103	233
7	-208	2	13	9	38	46	49
8	36	1	2	5	36	36	36
9	56	0	5	25	15	56	56
10	-33	2	1	29	8	94	224
11	-1392	1	2	3	2	5	150
12	1957	4	2	4	30	52	158
13	5993	1	9	10	7	24	82
14	-4147	4	1	7	35	44	222
15	-8635	3	1	14	16	1	103
16	4912	5	16	14	33	86	29
17	3568	5	15	3	1	12	227
18	-2992	4	0	15	1	56	92
19	4352	5	0	12	6	34	240
20	-336	0	4	5	33	45	178
21	-1408	6	3	18	27	116	134
22	1840	6	4	11	36	62	41
23	-4176	3	6	9	6	15	193
24	1488	4	9	0	12	91	203
25	-3040	5	3	29	35	8	44
26	-720	1	11	24	18	42	51
27	-1264	3	11	7	7	6	21
28	-1616	1	16	27	24	35	183
29	-464	5	12	1	28	44	50
30	-1424	4	4	2	11	100	118
31	-386	6	5	17	24	122	128
32	-864	4	3	4	38	25	164
33	-464	5	12	1	28	44	50
34	-464	5	12	1	28	44	50
35	-416	4	9	18	35	92	98
36	-288	6	1	22	40	93	226
37	-304	4	2	6	24	77	210
38	-192	4	12	25	13	62	65
39	-160	1	10	26	4	94	97
40	-96	2	6	28	27	31	161
41	-96	2	6	28	27	31	161
42	-16	5	1	15	25	111	241
43	-64	6	4	29	18	63	193
44	32	4	15	1	32	32	32
45	-48	1	3	14	34	79	209
46	48	6	14	17	7	48	48
47	-48	1	3	14	34	79	209
48	48	6	14	17	7	48	48
49	-32	3	2	30	9	95	225
50	32	4	15	1	32	32	32
51	-48	1	3	14	34	79	209
52	0	0	0	0	0	0	0
53	-48	1	3	14	34	79	209
54	0	0	0	0	0	0	0
55	-48	1	3	14	34	79	209
56	0	0	0	0	0	0	0
57	-48	1	3	14	34	79	209
58	0	0	0	0	0	0	0
59	32	4	15	1	32	32	32
60	0	0	0	0	0	0	0
61	0	0	0	0	0	0	0

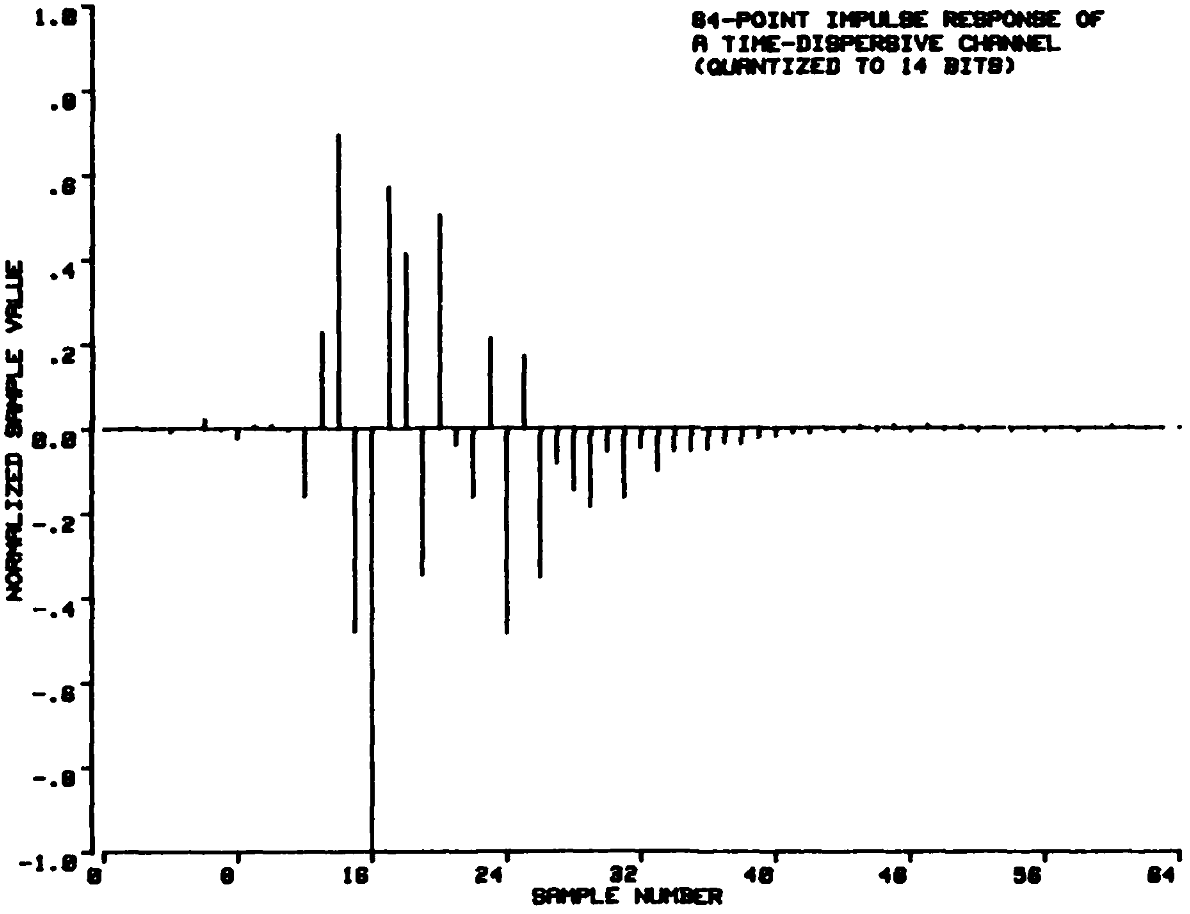


Figure 28. UNIT SAMPLE RESPONSE OF A TIME-DISPERSIVE CHANNEL (REPLICATING FILTER)

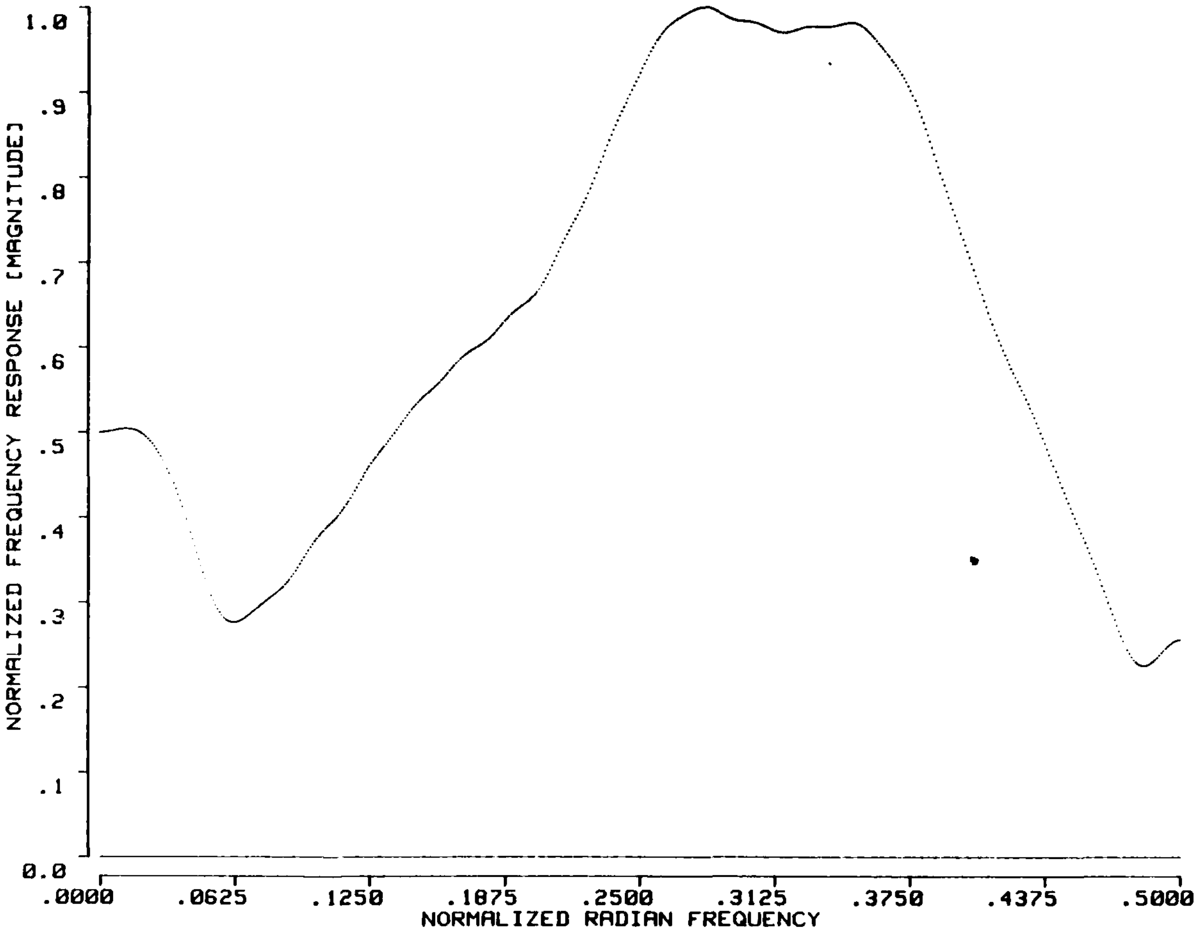


Figure 29. FREQUENCY RESPONSE OF A TIME-DISPERSIVE CHANNEL (REPLICATING FILTER)

M = 4936616111 will not be exceeded for input samples that are no larger in magnitude than 81336, supporting input quantization of 17 bits.

3.3.2 Inverse Channel Filter

The impulse response coefficients of the inverse channel filter, discussed in Section II as an equalizing filter, are presented in Table 11. The finite field decomposition corresponding to the diagram of Figure 27 is tabulated in Table 12. The impulse response of the inverse filter is plotted in Figure 30 with the corresponding frequency response shown in Figure 31. The filter was designed based on a criterion that minimized the rms value of the sidelobes in the compensated channel impulse response.

One can verify from the coefficients in Table 12 that the summed magnitudes of the coefficients is 285017. For a ring modulus of 4936616111, the input samples must be restricted in magnitude to 17320 or less if overflow is to be proscribed. The finite field design of the inverse filter in accordance with the diagram of Figure 27 will support 15-bit integer quantization of the input samples. In paıticular, it will support as input the unit impulse response of the channel filter with margin to spare.

3.3.3 Compensated Impulse Response

The impulse response of the compensated channel is shown in Figure 32. The calculation was made on the assumption that the channel output consisted of the unit impulse response of the replication filter which was used as the input to the inverse filter. The computations were carried out for the finite field decomposition of the inverse filter according to the diagram of Figure 27. The listing of a computer program that performed the simulation, written in BASIC

TABLE 11

UNIT SAMPLE RESPONSE OF THE INVERSE CHANNEL FILTER

k	h(k)	k	h(k)
1	1319	31	-854
2	-549	32	-4371
3	1126	33	-3231
4	-722	34	-12751
5	894	35	-4088
6	-854	36	-24116
7	687	37	-523
8	-849	38	-32767
9	579	39	6700
10	-778	40	-27813
11	392	41	14962
12	-745	42	-7025
13	228	43	16784
14	-556	44	16396
15	36	45	14428
16	-441	46	12585
17	-333	47	6271
18	-412	48	-26788
19	-854	49	-15061
20	-430	50	4902
21	-1334	51	7595
22	-367	52	1069
23	-1304	53	-397
24	-60	54	-1440
25	-632	55	455
26	267	56	151
27	258	57	-739
28	410	58	35
29	485	59	-580
30	-548	60	1382
		61	-309

TABLE 12

UNIT SAMPLE RESPONSE OF THE FINITE FIELD COMPONENTS

OF THE INVERSE CHANNEL FILTER

k	h(k)	MOD 7	MOD 17	MOD 31	MOD 41	MOD 127	MOD 257
1	1319	3	10	17	7	49	34
2	-549	4	12	9	25	86	222
3	1126	6	4	10	19	110	98
4	-722	6	9	22	16	40	49
5	894	5	10	26	33	5	123
6	-854	0	13	14	7	35	174
7	687	1	7	5	31	52	173
8	-849	5	1	19	12	40	179
9	579	5	1	21	5	71	65
10	-778	6	4	28	1	111	250
11	392	0	1	20	23	11	135
12	-745	4	3	30	34	17	26
13	228	4	7	11	23	101	228
14	-556	4	5	2	18	79	215
15	36	1	2	5	36	36	36
16	-441	0	1	24	10	67	73
17	-333	3	7	8	36	48	181
18	-412	1	13	22	39	96	102
19	-854	0	13	14	7	35	174
20	-430	4	12	4	21	78	84
21	-1334	3	9	30	19	63	208
22	-367	4	7	5	2	14	147
23	-1304	5	5	29	8	93	238
24	-60	3	8	2	22	67	197
25	-632	5	14	19	24	3	139
26	267	1	12	19	21	13	10
27	258	6	3	10	12	4	1
28	410	4	2	7	0	29	153
29	485	2	9	20	34	104	228
30	-548	5	13	10	26	87	223
31	-854	0	13	14	7	35	174
32	-4371	4	15	0	16	74	255
33	-3231	3	16	24	8	71	110
34	-12571	3	16	21	0	76	99
35	-4088	0	9	4	12	103	24
36	-24116	6	7	2	33	14	42
37	-523	2	4	4	10	112	248
38	-32767	0	9	0	33	126	129
39	6700	1	2	4	17	96	18
40	-27813	5	16	25	26	0	200
41	14962	3	2	20	38	103	56
42	-7025	3	13	12	27	87	171
43	16784	5	5	13	15	20	79
44	16396	2	8	28	37	13	205
45	14428	1	12	13	37	77	36
46	12585	6	5	30	39	12	249
47	6271	6	15	9	39	48	103
48	-26788	1	4	27	26	9	197
49	-15061	3	1	5	27	52	102
50	4902	2	6	4	23	76	19
51	7595	0	13	0	10	102	142
52	1069	5	15	15	3	53	41
53	-397	2	11	6	13	111	117
54	-1440	2	5	17	36	84	102
55	455	0	13	21	4	74	198
56	151	4	15	27	28	24	151
57	-739	3	9	5	40	23	32
58	35	0	1	4	35	35	35
59	-580	1	15	9	35	55	191
60	1382	3	5	18	29	112	97
61	-309	6	14	1	19	72	[illegible]

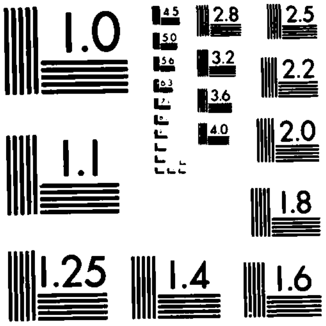
AD-A123 445 SAMPLE DATA PROCESSING(U) MITRE CORP BEDFORD MA 2/2
D O CARHOUN ET AL. AUG 82 MTR-8699 RADC-TR-82-227
F19628-81-C-0001

UNCLASSIFIED F/G 17/2 NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

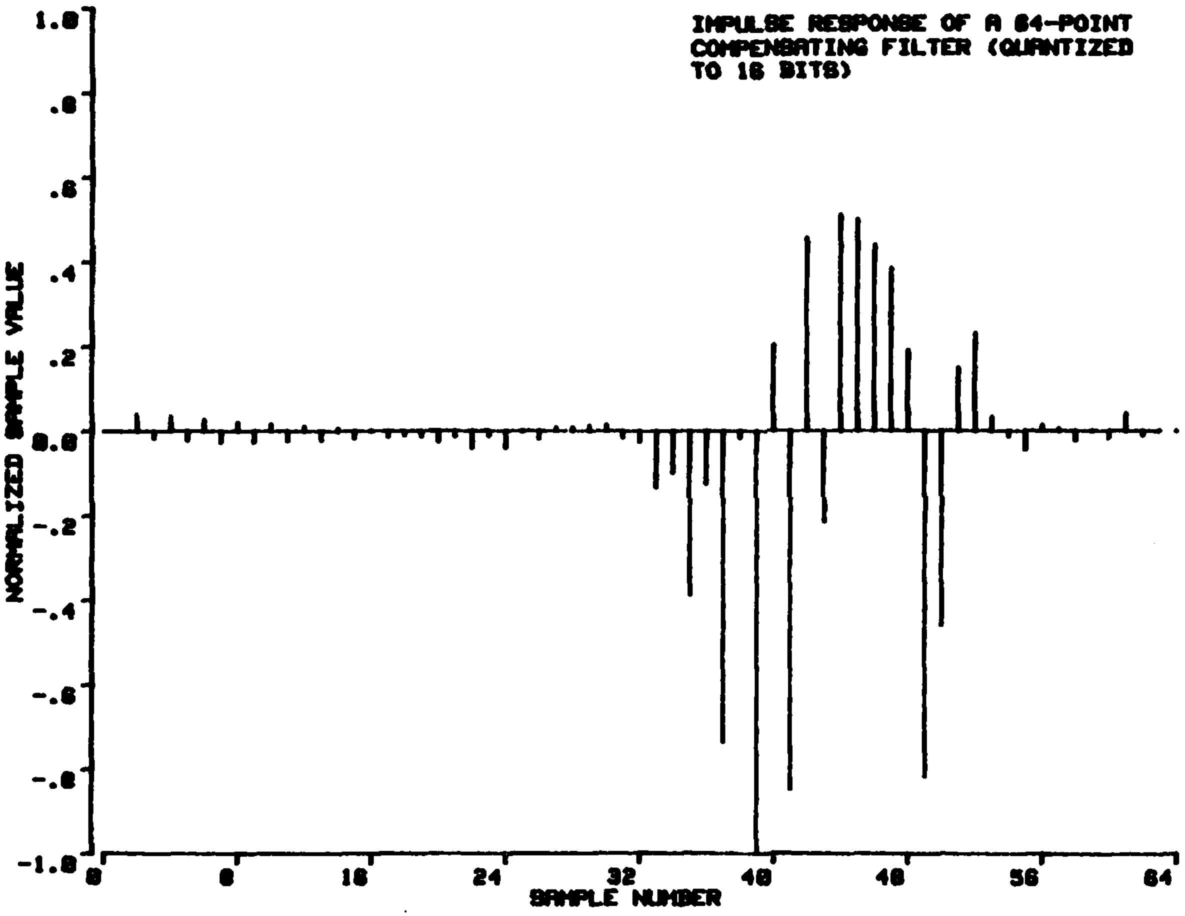


Figure 30. UNIT SAMPLE RESPONSE OF AN INVERSE EQUALIZING FILTER

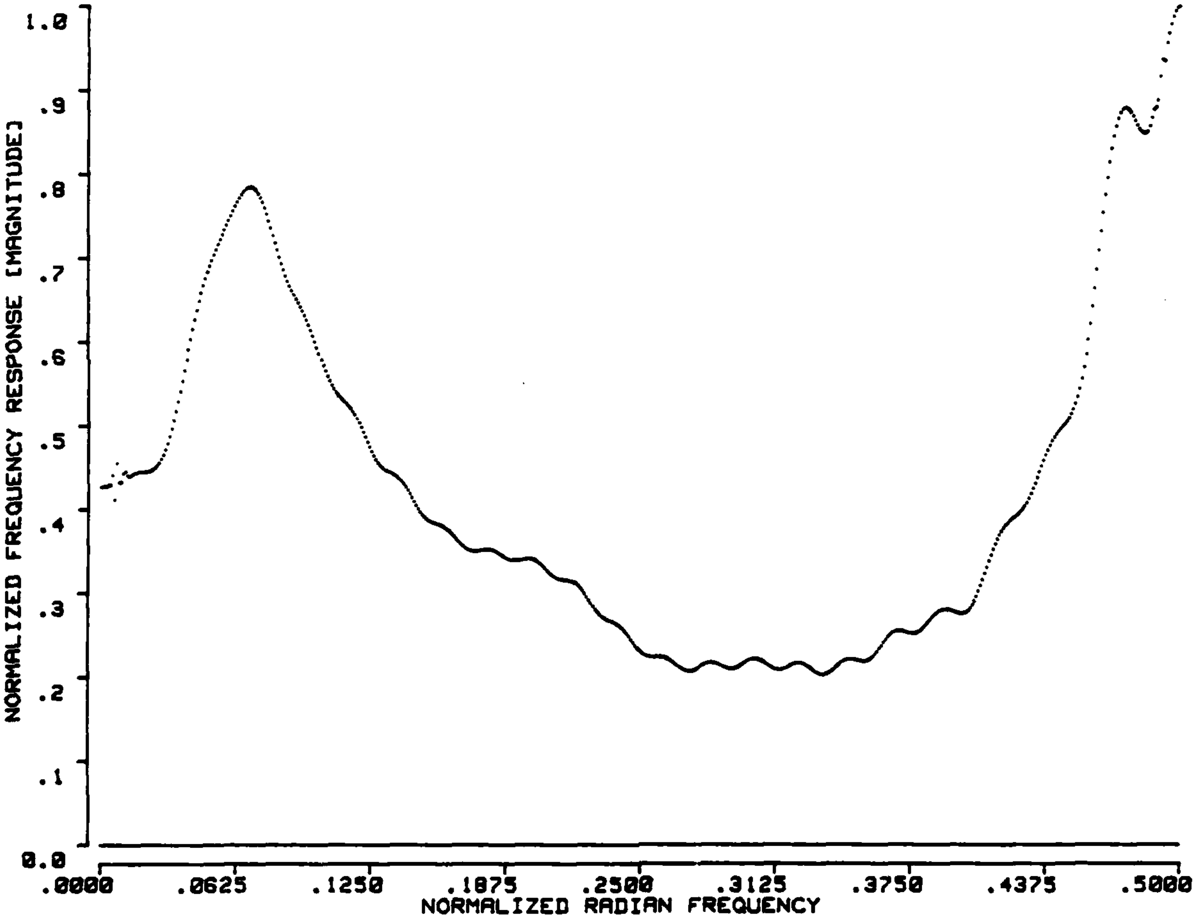


Figure 31. FREQUENCY RESPONSE OF AN INVERSE EQUALIZING FILTER

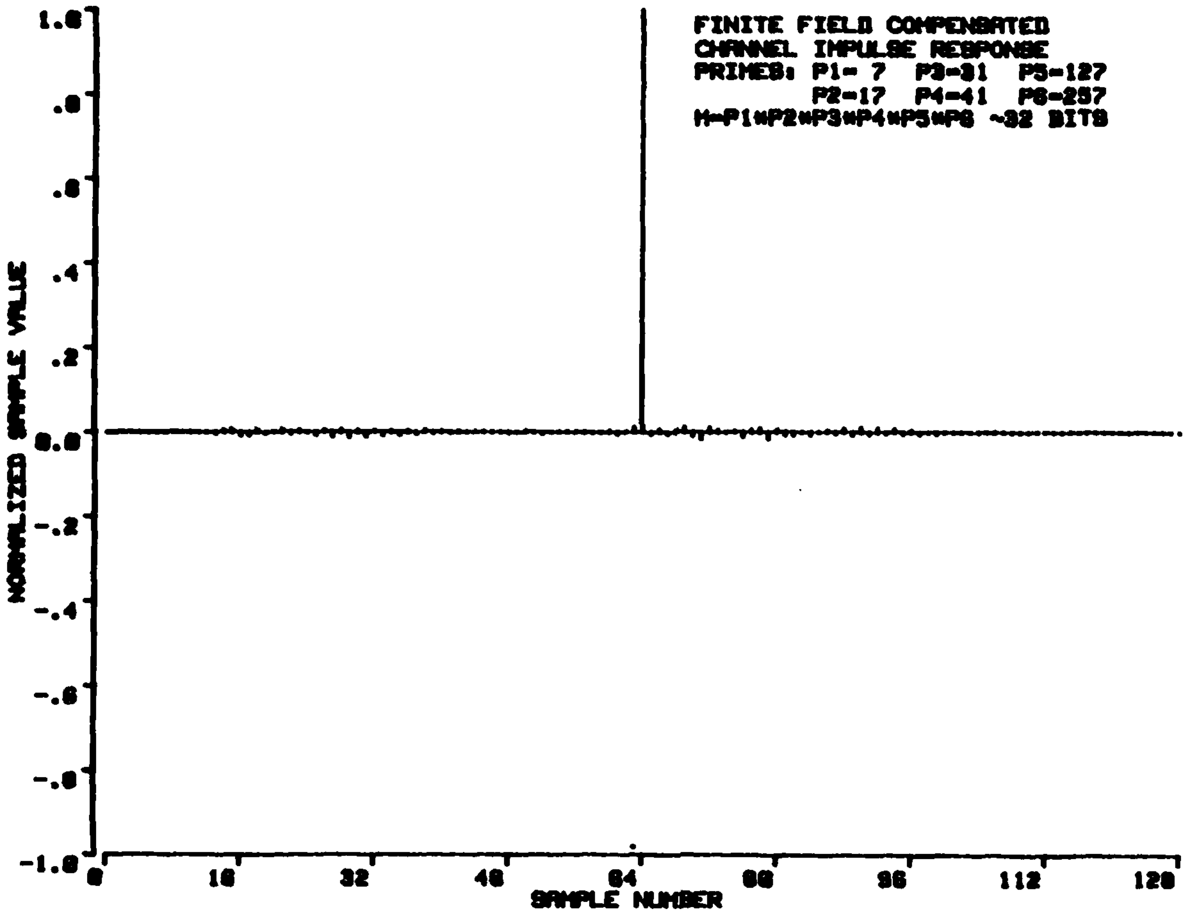


Figure 32. UNIT-SAMPLE RESPONSE OF THE COMPENSATED CHANNEL

to run on a Hewlett-Packard 9845 computer, is given in Appendix A. The result shown in Figure 32 is identical with a floating-point calculation using 12 digits, since there are no computational roundoff or overflow errors in the finite field simulation. The frequency response of the compensated channel is shown in Figure 33.

3.3.4 Cancellation of a PN Reference Sequence

Experiments were described in Section II that compared the performance of two methods of channel compensation for the purpose of stripping off a PN reference sequence from a transmitted pulse-amplitude-modulated signal. For the purpose of the experiment, a null signal was used; only the PN sequence was transmitted. Perfect cancellation at the receiver would have resulted in zero output corresponding to no signal. In practice, there was some residual error in each method caused by a combination of factors including computational roundoff, channel nonlinearity and imprecision in estimating the channel response.

Here we will describe the results of computer simulation of PN sequence cancellation for an assumed finite field implementation of the compensation filter. As received input, we will take the output of the simulated channel filter using as channel input a segment of a PN sequence that is long enough to cause intersymbol interference to be of concern. The compensation will presume the finite field implementation of Figure 27, but the received signal will be calculated using floating-point arithmetic. The finite field compensation/cancellation will be compared with conventional implementation using different levels of fixed-point arithmetic.

The PN sequence used for the computation, generated by computer, is shown in Tablr 13. The result of passing the sequence through the simulated time-dispersive channel is shown in Table 14. This

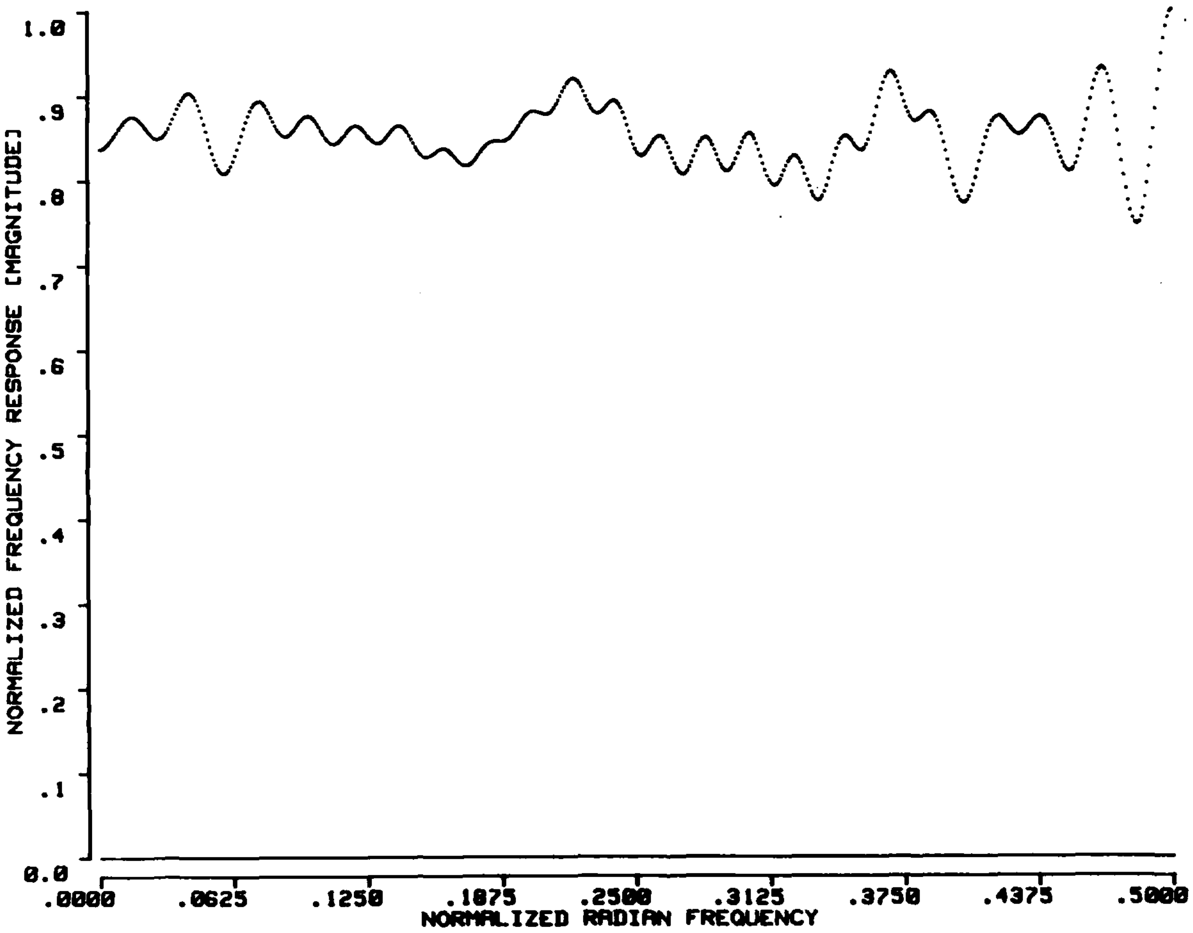


Figure 33. FREQUENCY RESPONSE OF THE COMPENSATED CHANNEL

TABLE 13

PSEUDO RANDOM (PN) REFERENCE SEQUENCE

k	x(k)
1	-83
2	-89
3	127
4	86
5	45
6	5
7	-36
8	-77
9	-117
10	98
11	57
12	16
13	-24
14	-65
15	-106
16	-56
17	-31
18	-5
19	21
20	46
21	72
22	97
23	123
24	-107
25	-46
26	-31
27	74
28	-77
29	28
30	-123
31	-18
32	87

TABLE 14

PN SEQUENCE TRANSMITTED THROUGH A
SIMULATED TIME-DISPERSIVE CHANNEL

k	S(k)	k	S(k)	k	S(k)
1	-1328	32	-519290	63	-8560
2	-2752	33	7803	64	31520
3	6169	34	111163	65	5984
4	9371	35	-119430	66	15360
5	-22349	36	-1451873	67	-2224
6	-20058	37	-139264	68	18176
7	40273	38	2340428	69	-5664
8	26845	39	995937	70	20336
9	-28384	40	-547100	71	-16368
10	-10826	41	348997	72	8576
11	122608	42	-312626	73	-28480
12	-56656	43	112702	74	4592
13	-865589	44	1360336	75	-19808
14	-27706	45	-732747	76	6640
15	1979408	46	-1388221	77	-14256
16	406400	47	270686	78	6928
17	-1868638	48	200586	79	-5472
18	-308575	49	-933546	80	15968
19	156824	50	1008628	81	4896
20	-927395	51	-628432	82	3488
21	682429	52	495870	83	-5504
22	1841635	53	7482	84	4432
23	663043	54	-120170	85	1888
24	-1315174	55	355612	86	-736
25	-864069	56	-238658	87	1760
26	349721	57	223244	88	-8112
27	-923536	58	-103942	89	-576
28	790838	59	56568	90	2784
29	657430	60	9798	91	0
30	-329884	61	-4844	92	0
31	339580	62	46002	93	0

table represents the received values from the channel. After being scaled and rounded to 15 bits, these values are used as the input signal to the compensator.

The residual error remaining after passing the received sequence through the inverse equalizing filter of Figure 30 and then subtracting a suitably delayed reference sequence with the relative gains adjusted for maximum cancellation is presented in Table 15. Also shown for comparison are results based on conventional implementation with several different levels of precision of fixed-point computation. For this example the mean-squared error increased from 2.59 to 4.75 for 8-bit rounding.

Similar results were obtained for the technique of channel compensation that employs channel replication. In this case the finite field implementation of the replicating filter's action on the stored reference sequence was compared in effect with implementation using fixed-point arithmetic. Of course, since the replicating filter is used both to model the channel and to prefilter the reference sequence, and since there is no error in finite field computation, there is no residual error. For this technique the major source of error would be imprecision in estimating the impulse response of the actual channel. The error caused by the fixed-point roundoff was slight, just as it was in the case of the inverse channel equalizer.

TABLE 15

RESIDUAL ERROR AFTER PN SEQUENCE CANCELLATION IN
AN INVERSE FILTER EQUALIZED CHANNEL

INDEX	FINITE FIELD COMPUTATION	ROUNDED TO 14 BITS	ROUNDED TO 11 BITS	ROUNDED TO 8 BITS
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0
8	0	0	0	0
9	0	0	0	0
10	0	0	0	0
11	0	0	0	0
12	0	0	0	0
13	-1	-1	-1	-1
14	0	0	0	1
15	2	2	2	2
16	0	0	0	1
17	-1	-1	-1	-1
18	0	0	0	0
19	-1	-1	-1	-2
20	-1	-1	-1	-1
21	1	1	1	1
22	2	2	2	3
23	1	1	1	1
24	-1	-1	-1	0
25	0	0	0	1
26	0	0	-1	0
27	-2	-2	-2	-1
28	2	2	2	2
29	-1	-1	-1	-1
30	1	1	1	0
31	0	0	0	0
32	0	0	-1	-1
33	0	0	0	-1
34	0	0	-1	0
35	0	0	0	0
36	-3	-3	-3	-2
37	1	1	1	2
38	2	2	2	4
39	1	2	1	2
40	0	0	0	2
41	2	2	2	3
42	-1	-1	-1	0
43	2	2	2	2
44	1	1	1	2
45	0	0	-1	1
46	-1	-1	-2	0
47	1	1	1	1
48	-2	-2	-2	0
49	-2	-2	-2	-2
50	2	2	2	8
51	-3	-3	-3	-5

TABLE 15

(CONTINUED)

INDEX	FINITE FIELD COMPUTATION	ROUNDED TO 14 BITS	ROUNDED TO 11 BITS	ROUNDED TO 8 BITS
52	3	3	3	6
53	-4	-4	-4	-6
54	3	3	3	5
55	-2	-2	-2	-3
56	3	3	3	4
57	0	0	-1	-3
58	2	2	1	4
59	-1	-1	-2	-2
60	2	1	2	3
61	-2	-2	-2	-6
62	0	0	0	2
63	2	2	2	1
64	1	1	1	2
65	-1	-1	-1	-2
66	0	0	0	2
67	-2	-2	-2	-3
68	-1	-1	-1	-3
69	3	3	3	6
70	3	3	2	2
71	-4	-4	-3	2
72	-2	-2	-2	-2
73	1	1	2	4
74	0	0	0	-3
75	1	1	1	0
76	0	0	0	-1
77	1	1	1	-3
78	-2	-2	-2	-4
79	-1	-1	-1	-2
80	2	2	1	-1
81	-1	-1	-1	1
82	2	2	2	1
83	0	0	0	2
84	-3	-3	-4	-4
85	1	1	1	1
86	0	0	0	1
87	1	1	1	1
88	2	2	2	5
89	-1	-1	-2	-1
90	-3	-3	-3	-4
91	-1	-1	-1	0
92	7	7	7	7
93	1	1	1	-1
94	-1	-1	-1	-1
95	1	1	0	1
96	-1	-1	-1	0
97	3	3	3	1
98	1	1	1	1
99	-4	-4	-4	-2
100	-4	-4	-4	-4
101	0	0	1	1
102	-2	-2	-2	-3

TABLE 15

(CONCLUDED)

INDEX	FINITE FIELD COMPUTATION	ROUNDED TO 14 BITS	ROUNDED TO 11 BITS	ROUNDED TO 8 BITS
103	0	0		-1
104	1	1		1
105	-2	-2		-3
106	3	3		2
107	1	1		2
108	0	0		-2
109	1	1		1
110	1	1	[illegible]	1
111	-1	-1	0	0
112	3	3	3	1
113	-3	-3	-3	-2
114	3	3	3	3
115	-3	-3	-3	-3
116	1	1	1	0
117	-3	-3	-3	-3
118	0	0	0	1
119	-2	-2	-2	-1
120	1	1	1	1
121	0	0	0	1
122	1	1	1	1
123	0	0	0	0
124	1	1	1	0
125	0	0	1	0
126	0	0	0	0
127	0	0	0	-1
128	-1	-1	-1	0
129	0	0	0	0
130	0	0	0	0
131	0	0	0	0
132	0	0	0	0
133	0	0	0	0
134	0	0	0	0
135	0	0	0	0
136	0	0	0	0
137	0	0	0	0
138	0	0	0	0
139	0	0	0	0
140	0	0	0	0
141	0	0	0	0
142	0	0	0	0
143	0	0	0	0
144	0	0	0	0
145	0	0	0	0
146	0	0	0	0
147	0	0	0	0
148	0	0	0	0
149	0	0	0	0
150	0	0	0	0
151	0	0	0	0
152	0	0	0	0
153	0	0	0	0
154	0	0	0	0

SECTION IV

CONCLUSIONS AND FUTURE WORK

4.1 GENERAL RESULTS

The Sampled Data Processing project in fiscal 1981 demonstrated by hardware simulation the feasibility of a potentially low-cost secure voice-bandwidth modem using digital processing with analog transmission that does not require source coding for bandwidth compression; nor is channel bandwidth expansion required. While originally intended for telephone line transmission, the techniques developed are readily adaptable to other channels where bandwidth must be conserved. The need for dispersive channel equalization was a major consideration in the development of the technique. Two methods were compared by simulation, and an unconventional method employing channel measurement and replication for compensation purposes showed somewhat greater performance for cancellation of a PN reference sequence used to mask the transmitted samples. Further engineering development is needed to make the concept a practical reality.

The secure voice-bandwidth model provided design examples of digital filters against which to compare the methods and results of finite field and conventionally designed digital filters. While the finite field methods are not critical to the practical implementation of the modem, they placed in evidence a number of advantages, particularly relevant to integrated circuit implementation, brought about by the natural parallelism of the processor's architecture when decomposed into the finite field representations. A principal advantage is that the complexity of hardware for direct convolution, found chiefly in the multipliers, is spread over a set of parallel processors in which the multipliers, using fewer bits of equivalent fixed-point precision, are simpler. Since the gate complexity of a multiplier array is proportional to the square of the word length, a significant advantage accrues. It impacts not only the physical complexity of gate count and layout, but also fosters high throughput. Parallelism also enhances testability.

4.1.1 Secure Voice-Bandwidth Modem

While the work performed this year demonstrated the essential feasibility of the modem concept, work remains to bring the development closer to reality. The first step would be to transmit and decode a voice signal masked by the encryption sequence and to perform all processing functions in real time. This would need to be accompanied by non-real-time simulation in order to optimize design parameters. Two methods of recovering the modulus component of the masking sequence were considered; one of them required partial equalization of the transmission channel while the other relies on high correlation between adjacent samples to resolve ambiguity. These methods need to be implemented and compared in real-time operation. All simulations were performed with a constant channel model that was measured accurately in advance. In practice, a means of periodically measuring the channel and adaptively updating the compensator needs to be incorporated. Finally, actual tests of a real-time operating modem on conventional voice-grade telephone lines remains.

4.1.2 Finite Field Processing

There are a number of avenues of further research and development of these techniques of finite field digital signal processing that are suggested by the results of this year's effort.

The design examples presented in Section III illustrated the use of the Chinese remainder theorem to decompose a large integer ring and then reconstruct the outputs of the set of parallel processors designed over the component fields. It is also possible to decompose the component processors into shorter segments based on a polynomial version of the Chinese remainder theorem. This is equivalent to decomposing the polynomial ring used for cyclic convolution into ideals generated by its modulus factors. The essential scheme of such a processor is shown in Figure 34. The properties and implementation of such a processor should be an important element of further research.

Although there are significant computational, performance, and hardware design advantages available for digital signal processing carried out in the mathematical framework of finite polynomial rings, one of the more compelling advantages to be explored is the possible ease of incorporating fault security in the hardware design and architecture of linear signal processing functions. This becomes particularly important when projecting the hardware design to VLSI levels involving minimum feature size of one micron or less and logic gate complexities of tens of thousands of gates per functional integrated circuit. Once the linear processors are mapped into finite fields, the use of algebraic error correcting codes to encode the finite states of the processor can be seriously considered. Investigation of the methods and performance of such error control techniques should be given emphasis in future work.

Testability of the finite field hardware design, when projected to very large scale integration, should be a major consideration. Testability of the processor structure, when committed to integrated circuit form, and sensitivity to soft errors will be related to the controllability and observability properties of the processor, viewed

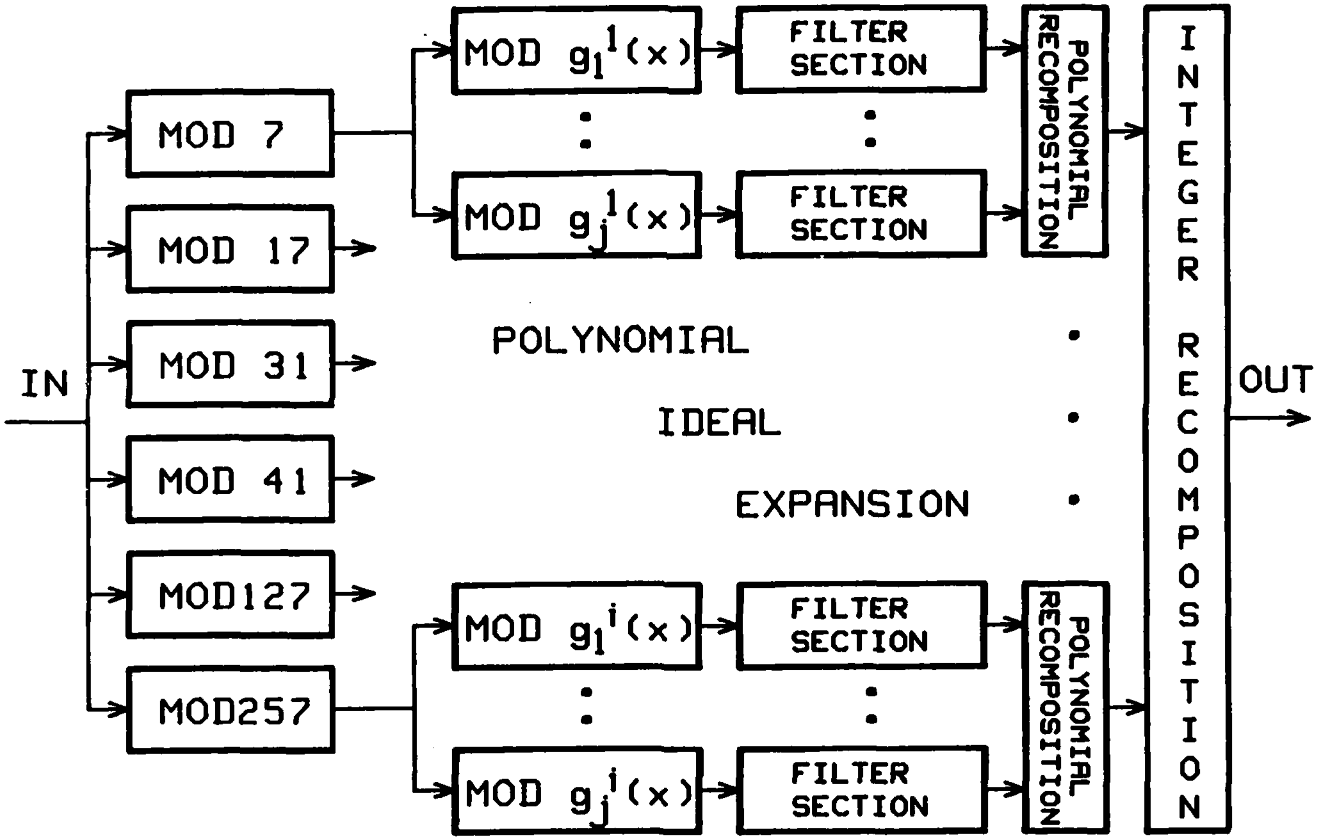


Figure 34. A DIGITAL FILTER DECOMPOSED INTO FINITE FIELD AND POLYNOMIAL IDEAL COMPONENTS

as a finite-state machine. If the structure is observable, the internal states can be calculated from the known input and observed response. If the structure is controllable, there is always some input that can drive the machine to any internal state in a finite number of steps. Canonical realizations of linear filters are always both completely observable and completely controllable, but the computational ease of determining an internal state or input control may depend on the particular form of realization. Non-canonical realizations, which may be attractive from an integrated architectural viewpoint, may sacrifice some desirable attributes of controllability or observability. These issues should be examined with a view to establishing criteria for testability and tolerance to soft errors.

The application to multi-dimensional processing and incorporation of hardware reconfigurability, or programmability, to fulfill a number of processing needs should be explored in future efforts.

LIST OF REFERENCES

1. R. J. Cosentino and S. J. Meehan, "Analog Secure Voice Modem" MITRE Technical Report MTR-8465, The MITRE Corporation, Bedford, MA, September 1981.

2. J. H. McClellan and C. M. Rader, Number Theory in Digital Signal Processing, Prentice Hall, 1979.

3. J. H. McClellan, T. W. Parks, L. R. Rabiner, "A Computer Program for Designing Optimum FIR Linear Phase Filters," IEEE Transactions on Audio and Electroacoustics, December 1973.

4. R. J. Cosentino and S. J. Meehan "Secure Voice - Bandwidth Modem", 1982 Carnahan Conference on Security Technology, May 1982.

APPENDIX A

BASIC PROGRAM FOR FINITE FIELD CONVOLUTION

This Appendix, included for the reader who is seriously interested in finite field signal processing, exhibits a BASIC software program that performs finite field convolution. The program listing below is accompanied by a printout of the example of Section 3.3 which convolved the impulse response of the replicating filter with that of the inverse channel filter to produce the impulse response of the compensated channel. The program was written to operate on a Hewlett-Packard 9845 C computer configured with 182K bytes of memory. Running time for the example is approximately five minutes. No attempt was made to make the program compact or to minimize running time or storage.

For convenience, the printout for the example is presented first, followed by the list of program statements. The sequence labeled A(*) is the channel response and B(*) is that of the inverse filter. Both are decomposed into finite field components as residues modulo P1 through P6 which are selected by the user. The sequence C(I) is the result of convolution in the six different fields. The values R1 through R6 are the scale factors needed in the Chinese remainder theorem reconstruction formula. The sequence Y(*) is the reconstruction of the finite field convolution into the integer ring Z_M. The peak value for the example occurs as Y(61).

APPENDIX A (CONTINUED)

```
***  FINITE FIELD CONVOLUTION  ***

SEQUENCE A(*) IS FROM FILE:REPLIC
DATA FILE `REPLIC' FROM:T15
LENGTH OF `REPLIC' IS 61

SEQUENCE B(*) IS FROM FILE:INVERS
DATA FILE `INVERS' FROM:T15
LENGTH OF `INVERS' IS 61

P1= 7
P2= 17
P3= 31
P4= 41
P5= 127
P6= 257
INTEGER RING MODULUS M= 4936616111
M=2^K, K= 32.2008753142

  A(*)    MOD  7    MOD 17    MOD 31    MOD 41    MOD127    MOD257

    16         2        16        16        16        16        16
    16         2        16        16        16        16        16
   -67         3         1        26        15        60       190
     0         0         0         0         0         0         0
   192         3         5         6        28        65       192
   -24         4        10         7        17       103       233
  -208         2        13         9        38        46        49
    36         1         2         5        36        36        36
    56         0         5        25        15        56        56
   -33         2         1        29         8        94       224
 -1392         1         2         3         2         5       150
  1957         4         2         4        30        52       158
  5993         1         9        10         7        24        82
 -4147         4         1         7        35        44       222
 -8635         3         1        14        16         1       103
  4912         5        16        14        33        86        29
  3568         5        15         3         1        12       227
 -2992         4         0        15         1        56        92
  4352         5         0        12         6        34       240
  -336         0         4         5        33        45       178
 -1408         6         3        18        27       116       134
  1840         6         4        11        36        62        41
 -4176         3         6         9         6        15       193
  1488         4         9         0        12        91       203
 -3040         5         3        29        35         8        44
  -720         1        11        24        18        42        51
 -1264         3        11         7         7         6        21
 -1616         1        16        27        24        35       183
  -464         5        12         1        28        44        50
 -1424         4         4         2        11       100       118
  -386         6         5        17        24       122       128
  -864         4         3         4        38        25       164
  -464         5        12         1        28        44        50
  -464         5        12         1        28        44        50
  -416         4         9        18        35        92        98
  -288         6         1        22        40        93       226
  -304         4         2         6        24        77       210
  -192         4        12        25        13        62        65
  -160         1        10        26         4        94        97
   -96         2         6        28        27        31       161
   -96         2         6        28        27        31       161
```

APPENDIX A (CONTINUED)

```
       16784         5         5        13        15        20        79
       16396         2         8        28        37        13       205
       14428         1        12        13        37        77        36
       12585         6         5        30        39        12       249
        6271         6        15         9        39        48       103
      -26788         1         4        27        26         9       197
      -15061         3         1         5        27        52       102
        4902         2         6         4        23        76        19
        7595         0        13         0        10       102       142
        1069         5        15        15         3        53        41
        -397         2        11         6        13       111       117
       -1440         2         5        17        36        84       102
         455         0        13        21         4        74       198
         151         4        15        27        28        24       151
        -739         3         9         5        40        23        32
          35         0         1         4        35        35        35
        -580         1        15         9        35        55       191
        1382         3         5        18        29       112        97
        -309         6        14         1        19        72       205

        C(I)    MOD  7    MOD 17    MOD 31    MOD 41    MOD127    MOD257

   C(    0)=         6         7        24        30        22        30
   C(    1)=         0        12        13        20         1       241
   C(    2)=         1        11         2        30       107        15
   C(    3)=         1        16         2        33        67        71
   C(    4)=         0         1        14        35        91       144
   C(    5)=         3        10        21        18        88       101
   C(    6)=         1         5        16        30        15       129
   C(    7)=         2        11         0        19       100        25
   C(    8)=         1        16        28        11        48       153
   C(    9)=         4        16        30         9        83       136
   C(   10)=         1         4        23         3        19         7
   C(   11)=         1         0        16         1        27       101
   C(   12)=         6        15         5        22        27       221
   C(   13)=         0        12         4        19        39         9
   C(   14)=         2         0        19         5        37        24
   C(   15)=         3         5         9        36        42        99
   C(   16)=         3        10        16        38        32        38
   C(   17)=         2         7         2        31        84       120
   C(   18)=         4        16        28         4        98        86
   C(   19)=         0         6        22        17        39        45
   C(   20)=         4        13        16        21       103       142
   C(   21)=         3        13        18         7        79       234
   C(   22)=         4        16         2         6       122       118
   C(   23)=         1         9        18        10        56       236
   C(   24)=         2         3         8        21        46       242
   C(   25)=         6         8        22        34        44         1
   C(   26)=         2        10        18        22        53       222
   C(   27)=         3         4         0        10        33       222
   C(   28)=         3         7         6        13        97        97
   C(   29)=         3        15        23        25        36         0
   C(   30)=         6        10        19        25        22       206
   C(   31)=         6        11        24        14       106       107
   C(   32)=         5        14        20        12       115        16
   C(   33)=         1         2         5        32        43       224
   C(   34)=         0        15        25        11        45       102
   C(   35)=         2         7         0        14        13        80
   C(   36)=         2         5         4        21        20       193
   C(   37)=         3         4        17        23       104        15
   C(   38)=         1        10         5         4        63        55
   C(   39)=         0         1        15        21        38       117
   C(   40)=         2         1        25         7       106        27
   C(   41)=         4         3        14        27       123       131
   C(   42)=         0        12        23        31       110       253
```

APPENDIX A (CONTINUED)

-16	5	1	15	25	111	241
-64	6	4	29	18	63	193
32	4	15	1	32	32	32
-48	1	3	14	34	79	209
48	6	14	17	7	48	48
-48	1	3	14	34	79	209
48	6	14	17	7	48	48
-32	3	2	30	9	95	225
32	4	15	1	32	32	32
-48	1	3	14	34	79	209
0	0	0	0	0	0	0
-48	1	3	14	34	79	209
0	0	0	0	0	0	0
-48	1	3	14	34	79	209
0	0	0	0	0	0	0
-48	1	3	14	34	79	209
0	0	0	0	0	0	0
32	4	15	1	32	32	32
0	0	0	0	0	0	0
0	0	0	0	0	0	0

B(*)	MOD 7	MOD 17	MOD 31	MOD 41	MOD127	MOD257
1319	3	10	17	7	49	34
-549	4	12	9	25	86	222
1126	6	4	10	19	110	98
-722	6	9	22	16	40	49
894	5	10	26	33	5	123
-854	0	13	14	7	35	174
687	1	7	5	31	52	173
-849	5	1	19	12	40	179
579	5	1	21	5	71	65
-778	6	4	28	1	111	250
392	0	1	20	23	11	135
-745	4	3	30	34	17	26
228	4	7	11	23	101	228
-556	4	5	2	18	79	215
36	1	2	5	36	36	36
-441	0	1	24	10	67	73
-333	3	7	8	36	48	181
-412	1	13	22	39	96	102
-854	0	13	14	7	35	174
-430	4	12	4	21	78	84
-1334	3	9	30	19	63	208
-367	4	7	5	2	14	147
-1304	5	5	29	8	93	238
-60	3	8	2	22	67	197
-632	5	14	19	24	3	139
267	1	12	19	21	13	10
258	6	3	10	12	4	1
410	4	2	7	0	29	153
485	2	9	20	34	104	228
-548	5	13	10	26	87	223
-854	0	13	14	7	35	174
-4371	4	15	0	16	74	255
-3231	3	16	24	8	71	110
-12751	3	16	21	0	76	99
-4088	0	9	4	12	103	24
-24116	6	7	2	33	14	42
-523	2	4	4	10	112	248
-32767	0	9	0	33	126	129
6700	1	2	4	17	96	18
-27813	5	16	25	26	0	200
14962	3	2	20	38	103	56
-7025	3	13	12	27	87	171

APPENDIX A (CONTINUED)

```
C(   43)=          3          2          5         18          0        113
C(   44)=          6          9         22          4        120        237
C(   45)=          1          6          3          4         30         32
C(   46)=          5          5          5          2         31        145
C(   47)=          1          6         13          0         77         81
C(   48)=          5         16         19         10         50         27
C(   49)=          1         13         20          9        125        117
C(   50)=          4         13         18         17         38        170
C(   51)=          2         16         20         23         23         93
C(   52)=          5          4         30         28         45        123
C(   53)=          3          3          6         32        102         44
C(   54)=          1          8         28          9         35        108
C(   55)=          4          6         26          7         66        116
C(   56)=          6         10         16         24         62        243
C(   57)=          6          5          1         29         92        205
C(   58)=          0          2         16         39         23         36
C(   59)=          4          1          5         13         20         42
C(   60)=          2         14         10         29         76         21
C(   61)=          5          3         29         34         54         42
C(   62)=          6          2         12         31         42        153
C(   63)=          2         14         29         20         87         53
C(   64)=          5         10         14         37         80         63
C(   65)=          0          4          4          1        110         42
C(   66)=          0         11         16         38         94         21
C(   67)=          4          7          0         20         75          7
C(   68)=          2         14         28         37         33        128
C(   69)=          6          0         23          3         51         90
C(   70)=          1         12         30         14        116         32
C(   71)=          5         15         10         37        103         25
C(   72)=          6          0         22          0        121         84
C(   73)=          5          0         17         24          9        146
C(   74)=          3         12         14         23        125        195
C(   75)=          6          3         21         38         19        185
C(   76)=          3          0          4         33         49        198
C(   77)=          3          6         22         25        125        227
C(   78)=          1         13         30         12         97         49
C(   79)=          5          5         27         31         95         37
C(   80)=          6          2         16         36         28        245
C(   81)=          1          8         11          2         89        121
C(   82)=          6          9         13         23         99        164
C(   83)=          3          3          3         16         35        186
C(   84)=          2          5         17         23         60         43
C(   85)=          3          9         25         13         72         78
C(   86)=          4          3          1         17        102         40
C(   87)=          2         16         23         16         97         19
C(   88)=          3          9         10         27         63        200
C(   89)=          4         10         20         14        106         19
C(   90)=          3          2          4         20         62        168
C(   91)=          2          0         24         24         62        122
C(   92)=          2          9         28         19         24         59
C(   93)=          4          2          4         11         38        160
C(   94)=          5         10         26         27         74         65
C(   95)=          0          0         19         39        110         44
C(   96)=          0         14         14          1        117        212
C(   97)=          4          5         12         17         78        122
C(   98)=          1         14          9          7        115         55
C(   99)=          0         14          6         29        119        226
C(  100)=          2          3         29         37         89         81
C(  101)=          5         14         23         32         12        249
C(  102)=          1         12          5         24         48        253
C(  103)=          6          2         27         16         99         84
C(  104)=          2          5          1         23         99        239
C(  105)=          4         10         30          4         43        139
C(  106)=          2          4         24         14          2        199
C(  107)=          5          0         26         19        117        234
C(  108)=          1          0          2         37         48        193
```

APPENDIX A (CONTINUED)

```
C( 109)=         4         3        10        13        15       186
C( 110)=         4        12        14        17        39       167
C( 111)=         1         2        21        16        68       217
C( 112)=         3         3        14         3       121       184
C( 113)=         5         5        25        38        62        38
C( 114)=         5         1        21         0       101         6
C( 115)=         3        13         8        15        62        62
C( 116)=         3        12        23         3        82       127
C( 117)=         5         7        18        26        28        20
C( 118)=         3         6         1        34        18       135
C( 119)=         0         0         0         0         0         0
C( 120)=         0         0         0         0         0         0

R1= 1410461746
R2= 4065448562
R3= 3184913620
R4= 3250942317
R5= 1438226741
R6= 1459855348
M= 4936616111
             Y(   0)=         21104
             Y(   1)=         12320
             Y(   2)=        -79141
             Y(   3)=         43247
             Y(   4)=        180558
             Y(   5)=        -88050
             Y(   6)=       -107554
             Y(   7)=         50654
             Y(   8)=        -41481
             Y(   9)=        -15284
             Y(  10)=      -1859388
             Y(  11)=       3344699
             Y(  12)=       5260240
             Y(  13)=      -5546051
             Y(  14)=      -5036148
             Y(  15)=       5152178
             Y(  16)=      -1991455
             Y(  17)=       -427271
             Y(  18)=       5321271
             Y(  19)=      -2742145
             Y(  20)=       3358618
             Y(  21)=       -611426
             Y(  22)=      -4351406
             Y(  23)=       2922326
             Y(  24)=      -8132780
             Y(  25)=       4594904
             Y(  26)=      -8979101
             Y(  27)=       2986819
             Y(  28)=      -7604790
             Y(  29)=       1961424
             Y(  30)=      -5944975
             Y(  31)=       2001880
             Y(  32)=      -4430407
             Y(  33)=       2737274
             Y(  34)=      -2289511
             Y(  35)=       3138564
             Y(  36)=      -1526901
             Y(  37)=       1748386
             Y(  38)=       -575882
             Y(  39)=       1505623
             Y(  40)=        354944
             Y(  41)=        673985
             Y(  42)=        303513
             Y(  43)=         44831
             Y(  44)=       -107703
```

APPENDIX A (CONTINUED)

```
Y( 45)=        -1414496
Y( 46)=         -140177
Y( 47)=         1005763
Y( 48)=         -803098
Y( 49)=        -2534160
Y( 50)=           21501
Y( 51)=          984146
Y( 52)=         1625391
Y( 53)=          385801
Y( 54)=         -550129
Y( 55)=         -831022
Y( 56)=         -232856
Y( 57)=         1840068
Y( 58)=        -3613127
Y( 59)=          857651
Y( 60)=         9198051
Y( 61)=       748380701
Y( 62)=        -3599646
Y( 63)=         4235156
Y( 64)=        -2892472
Y( 65)=         3835510
Y( 66)=         8579452
Y( 67)=        -6282615
Y( 68)=       -12225876
Y( 69)=         8189138
Y( 70)=         3597518
Y( 71)=         -743990
Y( 72)=         1330573
Y( 73)=        -8214859
Y( 74)=         3889119
Y( 75)=         6198254
Y( 76)=       -11773486
Y( 77)=         4985256
Y( 78)=        -4989606
Y( 79)=        -4901724
Y( 80)=          416842
Y( 81)=        -4778794
Y( 82)=          310106
Y( 83)=         2674528
Y( 84)=        -2827214
Y( 85)=         7114866
Y( 86)=        -4651146
Y( 87)=         8460202
Y( 88)=        -3416872
Y( 89)=         7188052
Y( 90)=        -2445958
Y( 91)=         4890832
Y( 92)=        -2461744
Y( 93)=         2492032
Y( 94)=        -3001952
Y( 95)=         -451248
Y( 96)=        -2825760
Y( 97)=        -1026336
Y( 98)=         -680224
Y( 99)=         -411488
Y(100)=          -53632
Y(101)=          828560
Y(102)=          613712
Y(103)=         1339568
Y(104)=          695424
Y(105)=        -1215728
Y(106)=         -712976
Y(107)=           88128
Y(108)=          357680
Y(109)=           18176
Y(110)=           43600
```

APPENDIX A (CONTINUED)

```
Y(111)=       -121344
Y(112)=         92704
Y(113)=        -63184
Y(114)=         19024
Y(115)=        -65216
Y(116)=         -3728
Y(117)=         44224
Y(118)=         -9888
Y(119)=             0
Y(120)=             0
Y(121)=             0
```

APPENDIX A (CONTINUED)

```
10     REM PROGRAM "CNVOLF" CONVOLVES TWO SEQUENCES OF ELEMENTS FROM GF(P).
20     REM THE SEQUENCES ARE ENTERED FROM SEPARATE FILES PREVIOUSLY STORED.
30     REM THE INPUT SEQUENCES MUST BE INTEGER SEQUENCES, NOT NECESSARILY FROM
           GF(P). THEY BELONG TO THE RING OF INTEGERS MOD M=P1*P2*P3*P4*P5*P6.
31     REM M MUST NOT EXCEED 12 DIGITS.
40     REM THE OUTPUT SEQUENCES ARE COMPUTED FOR 6 DIFFERENT PRIME MODULI
50     REM SPECIFIED BY THE USER.
60     REM THE OUTPUT SEQUENCES MAY BE STORED IN A FILE ARRAY TO BE READ
70     REM SEQUENTIALLY AND/OR MAY BE PRINTED.
80     REM IF STORED, THE ZEROTH OUTPUT TERM WILL ALWAYS BE THE MODULUS
90     REM ASSOCIATED WITH THE VALUES INDEXED FROM 1 TO N. THEREFORE IF
100    REM THE FILE IS READ SEQUENTIALLY AND THE CONVOLUTION VALUES ARE USED
110    REM IN A SUBSEQUENT PROGRAM WITH OPTION BASE ZERO, THEY SHOULD BE
120    REM DECREMENTED. INPUT VALUES ARE STORED IN THE SAME FILE NORMALLY
130    REM USING OPTION BASE ZERO.
140    REM BEGIN:
150    OPTION BASE 0
1000    INTEGER A(256),B(256),C(512),D(512),E(512),F(512),G(512),H(512)
1010    PRINTER IS 0
1020    PRINT "                      ***  FINITE FIELD CONVOLUTION  ***"
1021    PRINT
1022    PRINT
1030    LINPUT "ENTER FILE NAME OF FIRST SEQUENCE",A$
1040    PRINT "SEQUENCE A(*) IS FROM FILE:";A$
1050    LINPUT "ENTER NAME OF STORAGE DEVICE, eg.:T14 OR :T15",Ta$
1070    PRINT "DATA FILE `";A$;"´ FROM";Ta$
1080    ASSIGN A$&Ta$ TO #1
1090    INPUT "ENTER LENGTH OF SEQUENCE A(*)",L1
1100    PRINT "LENGTH OF `";A$;"´ IS";L1
1110    FOR I=1 TO 256          ! LOOP TO CLEAR INPUT REGISTERS
1120    A(I)=0
1130    B(I)=0
1140    NEXT I
1150    FOR I=0 TO L1-1
1160    READ #1;A(I)
1161      ON ERROR GOTO Illegal
1170    NEXT I
1180    PRINT
1190    PRINT
1200    REWIND Ta$
1210    BEEP
1220    LINPUT "ENTER FILE NAME OF SECOND SEQUENCE",B$
1230    PRINT "SEQUENCE B(*) IS FROM FILE:";B$
1240    LINPUT "ENTER NAME OF STORAGE DEVICE eg.:T14 OR:T15",Tb$
1250    PRINT "DATA FILE `";B$;"´ FROM";Tb$
1260    ASSIGN B$&Tb$ TO #2
1270    INPUT "ENTER LENGTH OF SEQUENCE B(*)",L2
1280    PRINT "LENGTH OF `";B$;"´ IS";L2
1300    FOR I=0 TO L2-1
1310    READ #2;B(I)
1311      ON ERROR GOTO Illegal
1320    NEXT I
1330    PRINT
1340    PRINT
1350    REWIND Tb$
1360    BEEP
1370    INPUT "ENTER PRIME P1",P1
1380    PRINT "P1=";P1
1390    INPUT "ENTER PRIME P2",P2
1400    PRINT "P2=";P2
1410    INPUT "ENTER PRIME P3",P3
```

```
1420    PRINT "P3=";P3
1430    INPUT "ENTER PRIME P4",P4
1440    PRINT "P4=";P4
1450    INPUT "ENTER PRIME P5",P5
1460    PRINT "P5=";P5
1470    INPUT "ENTER PRIME P6",P6
1480    PRINT "P6=";P6
1485    M=P1*P2*P3*P4*P5*P6
1490    PRINT "INTEGER RING MODULUS M=";M
1500    PRINT "M=2^K, K=";LOG(M)/LOG(2)
1510    PRINT
1520    PRINT
1530    FOR K=1 TO 512                ! INITIALIZE OUTPUT REGISTERS
1540    C(K)=0
1550    D(K)=0
1560    E(K)=0
1570    F(K)=0
1580    G(K)=0
1590    H(K)=0
1600    NEXT K
1610    C(0)=P1
1620    D(0)=P2
1630    E(0)=P3
1640    F(0)=P4
1650    G(0)=P5
1660    H(0)=P6
1670    BEEP
1680    LINPUT "PRINT INPUT SEQUENCES ?",Q$
1690    IF Q$<>"Y" THEN 2000
1700    IMAGE XXXAAAAXXX,XAAA,DDDXXX,XAAA,DDDXXX,XAAA,DDDXXX,XAAA,DDDXXX,XAAA,DDD
XXX,XAAA,DDDXXX/
1710    PRINT USING 1700;"A(*)","MOD",P1,"MOD",P2,"MOD",P3,"MOD",P4,"MOD",P5,"MOD
",P6
1720    IMAGE DDDDDDDXXX,XDDDDDDXXX,XDDDDDDXXX,XDDDDDDXXX,XDDDDDDXXX,XDDDDDDXXX,X
DDDDDDXXX
1730    FOR I=O TO L1-1
1740    PRINT USING 1720;A(I),A(I) MOD P1,A(I) MOD P2,A(I) MOD P3,A(I) MOD P4,A(I
) MOD P5,A(I) MOD P6
1750    NEXT I
1751    PRINT
1752    PRINT
1760    PRINT USING 1700;"B(*)","MOD",P1,"MOD",P2,"MOD",P3,"MOD",P4,"MOD",P5,"MOD
",P6
1761    FOR I=0 TO L2-1
1770    PRINT USING 1720;B(I),B(I) MOD P1,B(I) MOD P2,B(I) MOD P3,B(I) MOD P4,B(I
) MOD P5,B(I) MOD P6
1780    NEXT I
1790    PRINT
1800    PRINT
1900    DISP "COMPUTING CONVOLUTION"
2000    C(1)=A(0)*B(0) MOD P1          ! CONVOLUTION BEGINS
2010    FOR K=2 TO L1+L2-1
2020    Z=0
2030      FOR J=0 TO K-1
2040      W=A(J)*B(K-1-J) MOD P1
2050      Z=(Z+W) MOD P1
2060      NEXT J
2070    C(K)=Z
2080    NEXT K
2090    DISP "CONVOLUTION MOD P1 DONE"
2100    D(1)=A(0)*B(0) MOD P2
2110    FOR K=2 TO L1+L2-1
2120    Z=0
2130      FOR J=0 TO K-1
2140      W=A(J)*B(K-1-J) MOD P2
2150      Z=(Z+W) MOD P2
```

```
2160      NEXT J
2170    D(K)=Z
2180    NEXT K
2190    DISP "CONVOLUTION MOD P2 DONE"
2200    E(1)=A(0)*B(0) MOD P3
2210    FOR K=2 TO L1+L2-1
2220    Z=0
2230      FOR J=0 TO K-1
2240      W=A(J)*B(K-1-J) MOD P3
2250      Z=(Z+W) MOD P3
2260      NEXT J
2270    E(K)=Z
2280    NEXT K
2290    DISP "CONVOLUTION MOD P3 DONE"
2300    F(1)=A(0)*B(0) MOD P4
2310    FOR K=2 TO L1+L2-1
2320    Z=0
2330      FOR J=0 TO K-1
2340      W=A(J)*B(K-1-J) MOD P4
2350      Z=(Z+W) MOD P4
2360      NEXT J
2370    F(K)=Z
2380    NEXT K
2390    DISP "CONVOLUTION MOD P4 DONE"
2400    G(1)=A(0)*B(0) MOD P5
2410    FOR K=2 TO L1+L2-1
2420    Z=0
2430      FOR J=0 TO K-1
2440      W=A(J)*B(K-1-J) MOD P5
2450      Z=(Z+W) MOD P5
2460      NEXT J
2470    G(K)=Z
2480    NEXT K
2490   DISP "CONVOLUTION MOD P5 DONE"
2500    H(1)=A(0)*B(0) MOD P6
2510    FOR K=2 TO L1+L2-1
2520    Z=0
2530      FOR J=0 TO K-1
2540      W=A(J)*B(K-1-J) MOD P6
2550      Z=(Z+W) MOD P6
2560      NEXT J
2570    H(K)=Z
2580    NEXT K                         ! CONVOLUTION ENDS
2590   DISP "CONVOLUTION MOD P6 DONE"
2600    BEEP
2610    BEEP
3000    LINPUT "PRINT RESULTS OF CONVOLUTION ?",R$
3010    IF R$<>"Y" THEN 4000
3020    IMAGE XXXAAAAXXX,XAAA,DDDXXX,XAAA,DDDXXX,XAAA,DDDXXX,XAAA,DDDXXX,XAAA,DDD
XXX,XAAA,DDDXXX/
3030    PRINT USING 3020;"C(I)","MOD",P1,"MOD",P2,"MOD",P3,"MOD",P4,"MOD",P5,"MOD
",P6
3040    IMAGE AAA,DDD,AAXXX,XDDDDDDXXX,XDDDDDDXXX,XDDDDDDXXX,XDDDDDDXXX,XDDDDDDXX
X,XDDDDDDXXX
3060    FOR K=1 TO L1+L2-1
3070    PRINT USING 3040;"C(",K-1,")=",C(K),D(K),E(K),F(K),G(K),H(K)
3080    NEXT K
3090    PRINT
3100    PRINT
3110    BEEP
4000    LINPUT "SAVE RESULTS IN DATA FILE ?",W$
4010    IF W$<>"Y" THEN 7500
4020    LINPUT "ENTER NAME OF NEW FILE",N$
4030    LINPUT "ENTER NAME OF STORAGE DEVICE, eg:T14 OR :T15",V$
4040    PRINT "RESULTS IN FILE `";N$;"'OF LENGTH";L1+L2
4050    PRINT "RECORDED ON";V$
```

```
4060    PRINT
4070    PRINT
4080    CREATE N$&V$,512,32
4090    ON ERROR GOTO Illegal
4100    ASSIGN N$&V$ TO #3
4110    PRINT #3;A(*),B(*),C(*),D(*),E(*),F(*),G(*),H(*)
4120    ON ERROR GOTO Illegal
4130    REWIND V$
4140    PRINT "RESULTS STORED IN FILE";N$;"(ON";V$;")"
4150    PRINT
4160    PRINT
4170    BEEP
5000    LINPUT " DO YOU WISH TO EXAMINE STORED DATA ?",E$
5010    IF E$<>"Y" THEN 10000
5020    REM !!  LINPUT "ENTER NAME OF DATA FILE",N$
5030    REM !!  LINPUT "ENTER NAME OF STORAGE DEVICE, eg :T14 OR:T15",V$
5040    ASSIGN N$&V$ TO #3
5050    READ #3;A(*),B(*),C(*),D(*),E(*),F(*),G(*),H(*)
5060    ON ERROR GOTO Illegal
5070    REWIND V$
5080    PRINT "                    *****DATA RETRIEVED FROM FILE FOLLOWS****"
5090    PRINT
5100    PRINT "P1=";C(0)
5110    PRINT "P2=";D(0)
5120    PRINT "P3=";E(0)
5130    PRINT "P4=";F(0)
5140    PRINT "P5=";G(0)
5150    PRINT "P6=";H(0)
5151    P1=C(0)
5152    P2=D(0)
5153    P3=E(0)
5154    P4=F(0)
5155    P5=G(0)
5156    P6=H(0)
5160    PRINT
5170    PRINT
6700    IMAGE XXXAAAAXXX,XAAA,DDDXXX,XAAA,DDDXXX,XAAA,DDDXXX,XAAA,DDDXXX,XAAA,DDD
XXX,XAAA,DDDXXX/
6710    PRINT USING 6700;"A(*)","MOD",P1,"MOD",P2,"MOD",P3,"MOD",P4,"MOD",P5,"MOD
",P6
6720    IMAGE DDDDDDDXXX,XDDDDDDXXX,XDDDDDDXXX,XDDDDDDXXX,XDDDDDDXXX,XDDDDDDXXX,X
DDDDDDXXX
6730    FOR I=0 TO L1-1
6740    PRINT USING 6720;A(I),A(I) MOD P1,A(I) MOD P2,A(I) MOD P3,A(I) MOD P4,A(I
) MOD P5,A(I) MOD P6
6750    NEXT I
6760    PRINT USING 6700;"B(*)","MOD",P1,"MOD",P2,"MOD",P3,"MOD",P4,"MOD",P5,"MOD
",P6
6761    FOR I=0 TO L2-1
6770    PRINT USING 6720;B(I),B(I) MOD P1,B(I) MOD P2,B(I) MOD P3,B(I) MOD P4,B(I
) MOD P5,B(I) MOD P6
6780    NEXT I
6790    PRINT
6800    PRINT
7020    IMAGE XXXAAAAXXX,XAAA,DDDXXX,XAAA,DDDXXX,XAAA,DDDXXX,XAAA,DDDXXX,XAAA,DDD
XXX,XAAA,DDDXXX/
7030    PRINT USING 7020;"C(I)","MOD",P1,"MOD",P2,"MOD",P3,"MOD",P4,"MOD",P5,"MOD
",P6
7040    IMAGE AAA,DDD,AAXXX,XDDDDDDXXX,XDDDDDDXXX,XDDDDDDXXX,XDDDDDDXXX,XDDDDDDXX
X,XDDDDDDXXX
7060    FOR K=1 TO L1+L2-1
7070    PRINT USING 7040;"C(",K-1,")=",C(K),D(K),E(K),F(K),G(K),H(K)
7080    NEXT K
7090    PRINT
7100    PRINT
7110    BEEP
```

APPENDIX A (CONTINUED)

```
7500    LINPUT "DO YOU WISH TO RECONSTRUCT RESULTS IN Z/MOD M",M$
7510    IF M$<>"Y" THEN 10000
7520    REM Chinese_remainder
7530    REM !! ALGORITHM:(1) FIND LEAST K FOR WHICH M1*K MOD P1=1,(2)COMPUTE
                         M1*K MOD M=R1.
7540    M=P1*P2*P3*P4*P5*P6
7550    M1=M/P1
7570    FOR K=1 TO M
7574    U=K*M1 MOD P1
7586    IF U=1 THEN 7600
7596    NEXT K
7600    R1=K*M1 MOD M
7650    M2=M/P2
7670    FOR K=1 TO M
7684    U=K*M2 MOD P2
7686    IF U=1 THEN 7700
7696    NEXT K
7700    R2=K*M2 MOD M
7750    M3=M/P3
7770    FOR K=1 TO M
7772    U=K*M3 MOD P3
7786    IF U=1 THEN 7800
7796    NEXT K
7800    R3=K*M3 MOD M
7850    M4=M/P4
7870    FOR K=1 TO M
7872    U=K*M4 MOD P4
7886    IF U=1 THEN 7900
7896    NEXT K
7900    R4=K*M4 MOD M
7950    M5=M/P5
7970    FOR K=1 TO M
7972    U=K*M5 MOD P5
7986    IF U=1 THEN 8000
7996    NEXT K
8000    R5=K*M5 MOD M
8050    M6=M/P6
8070    FOR K=1 TO M
8072    U=K*M6 MOD P6
8086    IF U=1 THEN 8100
8096    NEXT K
8100    R6=K*M6 MOD M
8101    PRINT "R1=";R1
8102    PRINT "R2=";R2
8103    PRINT "R3=";R3
8104    PRINT "R4=";R4
8105    PRINT "R5=";R5
8106    PRINT "R6=";R6
8107    PRINT "M=";M
8200    DIM O(512),P(512),Q(512),W(512),X(512),Y(512),Z(512)
8210    FOR I=0 TO L1+L2-2
8220    O(I)=C(I+1)*R1 MOD M
8230    P(I)=D(I+1)*R2 MOD M
8240    Q(I)=E(I+1)*R3 MOD M
8250    W(I)=F(I+1)*R4 MOD M
8260    X(I)=G(I+1)*R5 MOD M
8270    Z(I)=H(I+1)*R6 MOD M
8280    Y1=(O(I)+P(I)) MOD M
8281    Y2=(Y1+Q(I)) MOD M
8282    Y3=(Y2+W(I)) MOD M
8283    Y4=(Y3+X(I)) MOD M
8284    Y5=(Y4+Z(I)) MOD M
8285    Y(I)=Y5
8286    IF Y5<=(M-1)/2 THEN 8291
8287    Y(I)=Y5-M
8291    NEXT I
```

APPENDIX A (CONCLUDED)

```
8300   BEEP
8310   LINPUT "PRINT FINAL RESULT OF FINITE FIELD CONVOLUTION ?",Y$
8320   IF Y$<>"Y" THEN 8400
8330   IMAGE XXXXXXXXXXXXXXAA,DDD,AA,XXXDDDDDDDDDDDDXXXXX
8340   FOR I=0 TO L1+L2-1
8350   PRINT USING 8330;"Y(",I,")=",Y(I)
8360   NEXT I
8370   PRINT
8380   PRINT
8390   BEEP
8400   LINPUT "STORE FINAL RESULT IN DATA FILE ?",D$
8410   IF D$<>"Y" THEN 8300
8420   LINPUT "ENTER NAME OF NEW DATA FILE",G$
8430   LINPUT "ENTER NAME OF STORAGE DEVICE, eg :T14 OR :T15",H$
8440   CREATE G$&H$,128
8450   ON ERROR GOTO Illegal
8460   ASSIGN G$&H$ TO #4
8470   ON ERROR GOTO Illegal
8480   PRINT #4;Y(*)
8490   REWIND H$
8491   PRINT "RECONSTRUCTED DATA IN FILE:";G$;"OF LENGTH:";L2+L1;"RECORDED ON";H
$
8500 Illegal:   BEEP
8510            Error_type=ERRN
8520            Error_line=ERRL
8530            OFF ERROR
8540            DISP "ERROR";Error_type;"IN LINE";Errorline
8550            STOP
8560            RETURN
10000  DISP "END OF PROGRAM"
10010  PRINT
10020  PRINT
10030  PRINT "END OF PROGRAM"
10040  BEEP
10050  BEEP
10060  BEEP
10070  BEEP
10080  BEEP
10090  BEEP
10100  END
```

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C^{3}I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.